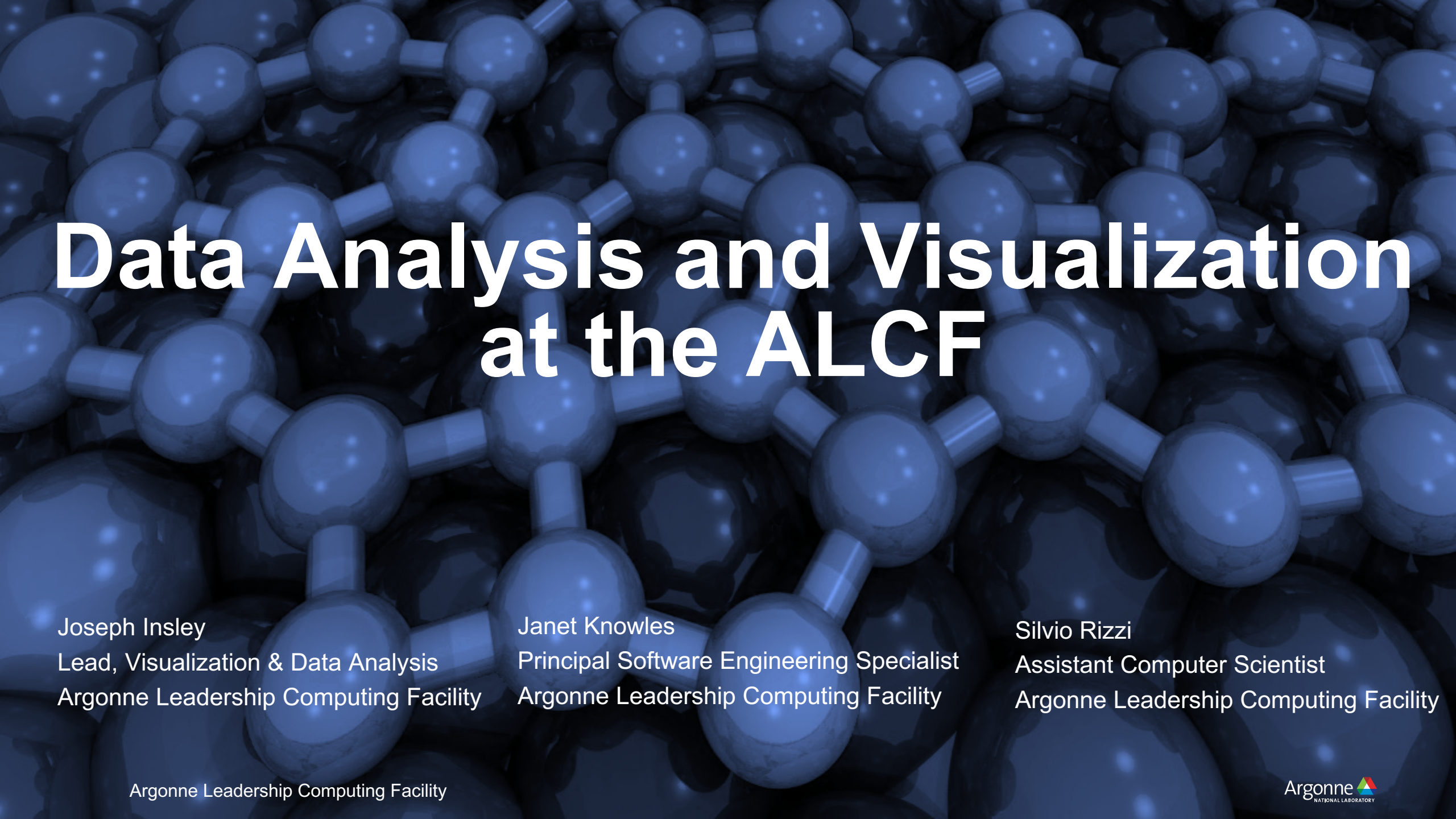


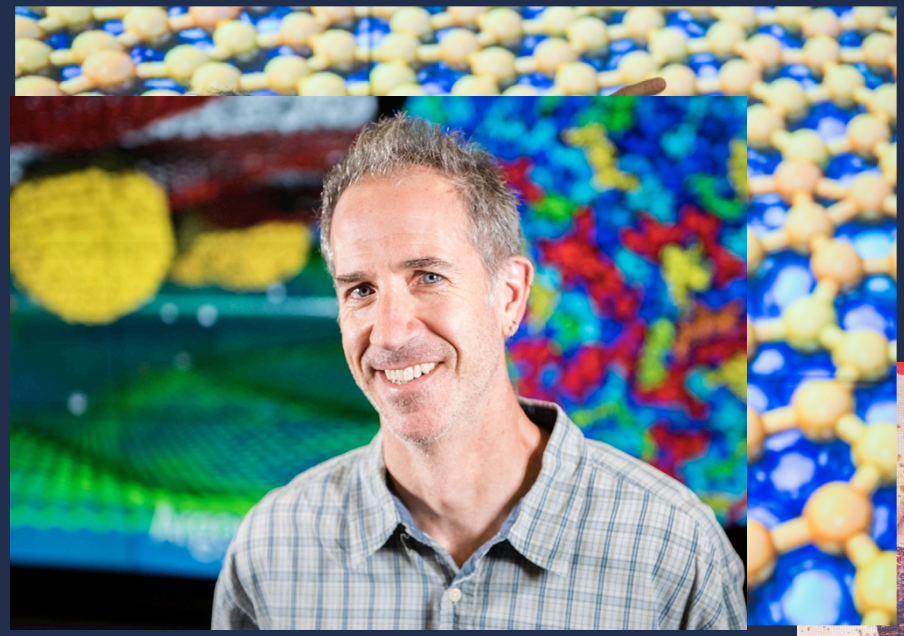
Data Analysis and Visualization at the ALCF

Joseph Insley
Lead, Visualization & Data Analysis
Argonne Leadership Computing Facility

Janet Knowles
Principal Software Engineering Specialist
Argonne Leadership Computing Facility

Silvio Rizzi
Assistant Computer Scientist
Argonne Leadership Computing Facility

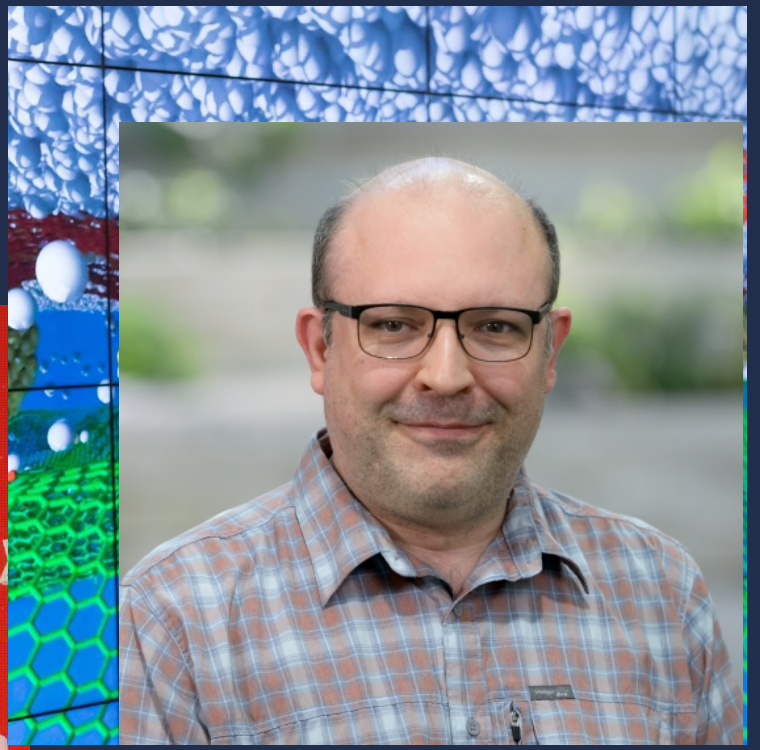
ALCF Visualization Group



Joe Insley

Janet Knowles



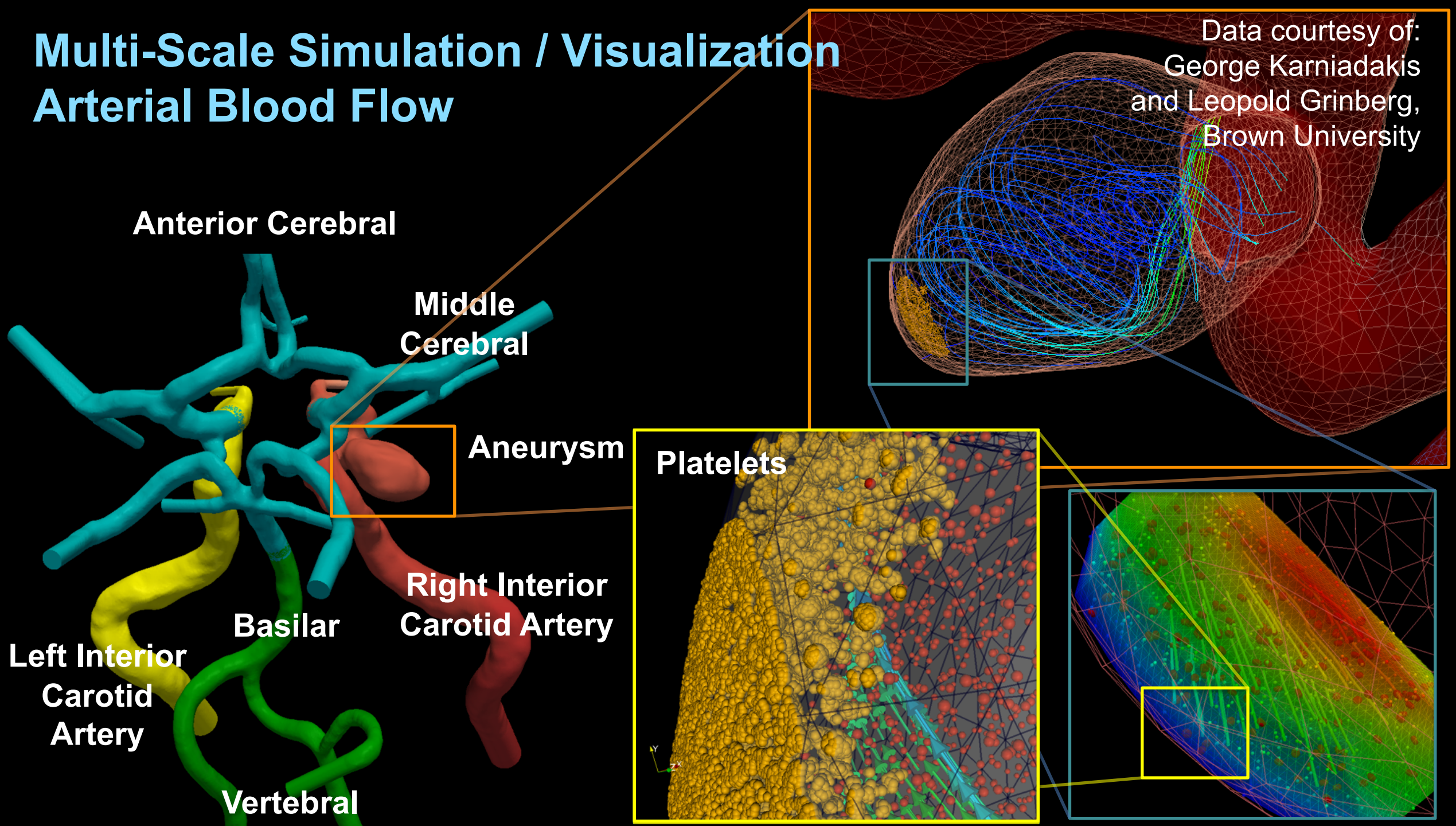


Silvio Rizzi

Here’s the plan…

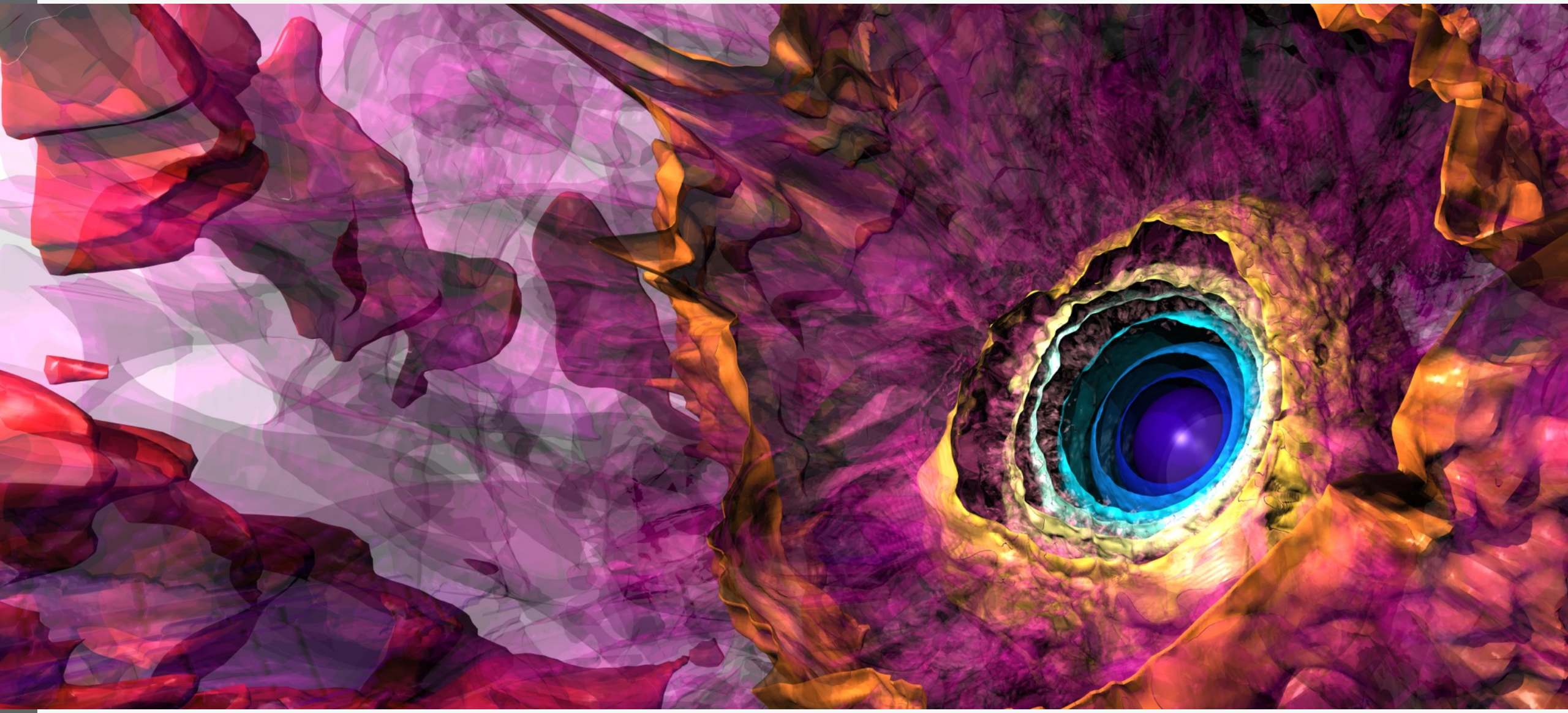
- Examples of visualizations
- Visualization resources
- Visualization tools and formats
- Data representations
- Visualization for debugging
- In Situ Visualization and Analysis

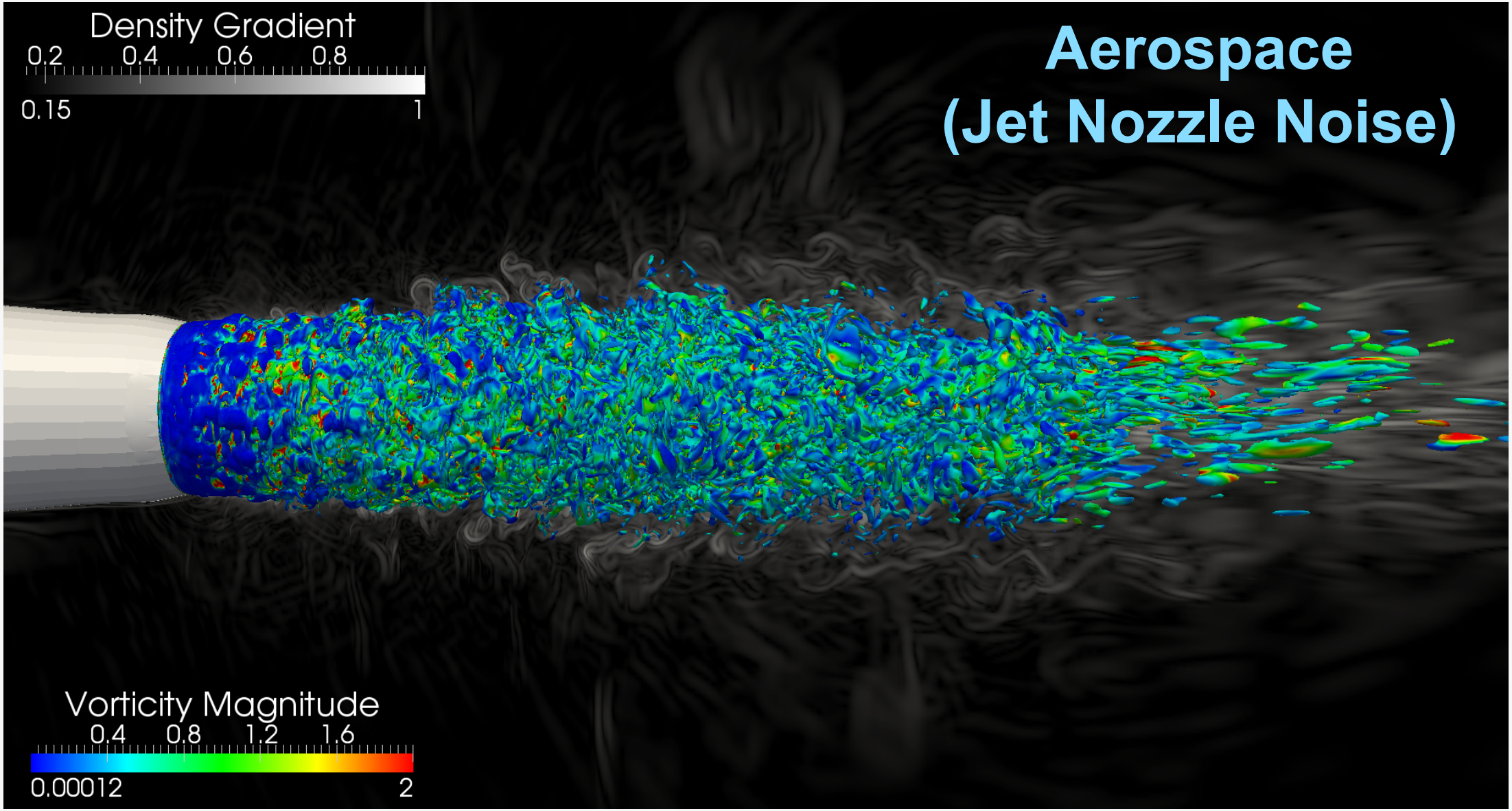
Multi-Scale Simulation / Visualization
Arterial Blood Flow



Physics

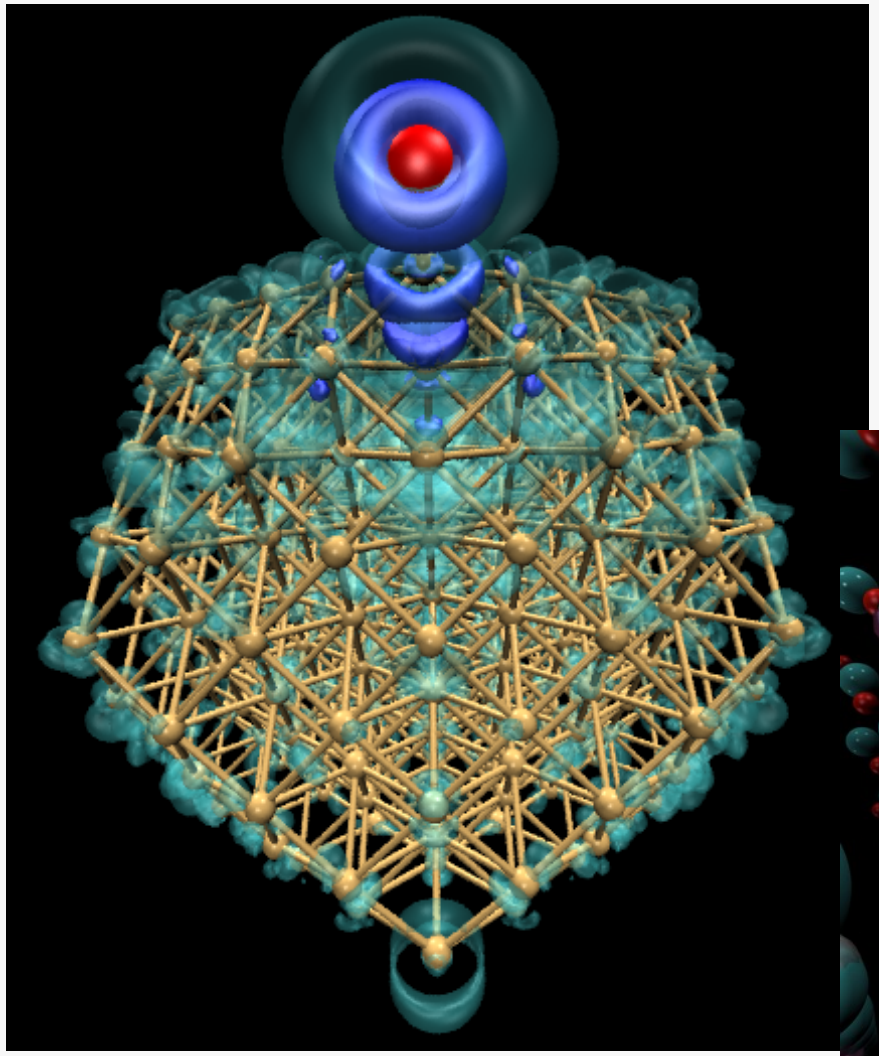
Data courtesy of: Lars Bildsten and Yan-Fei Jiang, University of California at Santa Barbara





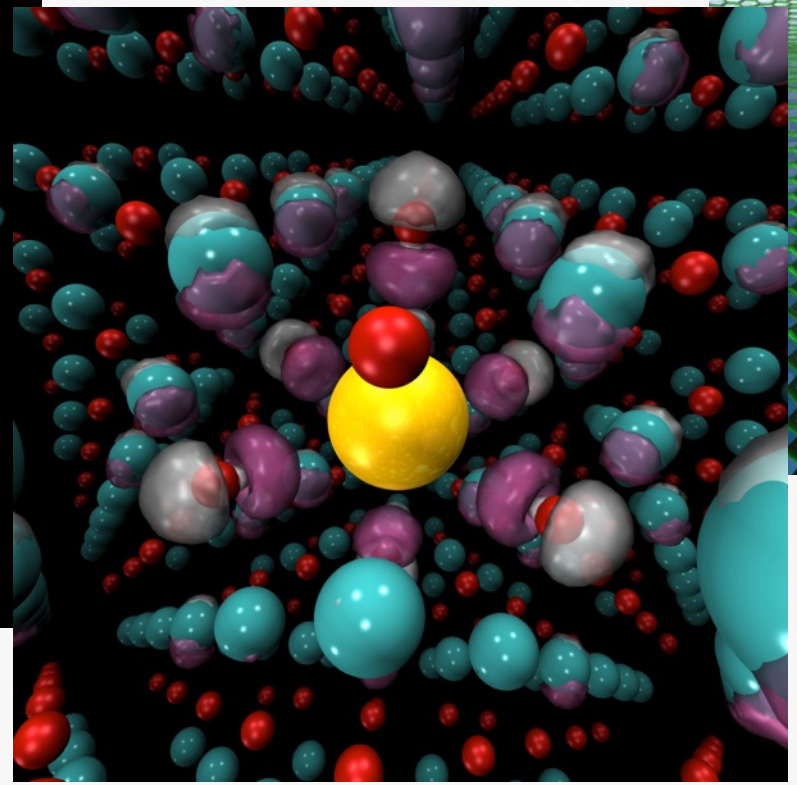
Data courtesy of: Anurag Gupta and Umesh Paliath, General Electric Global Research

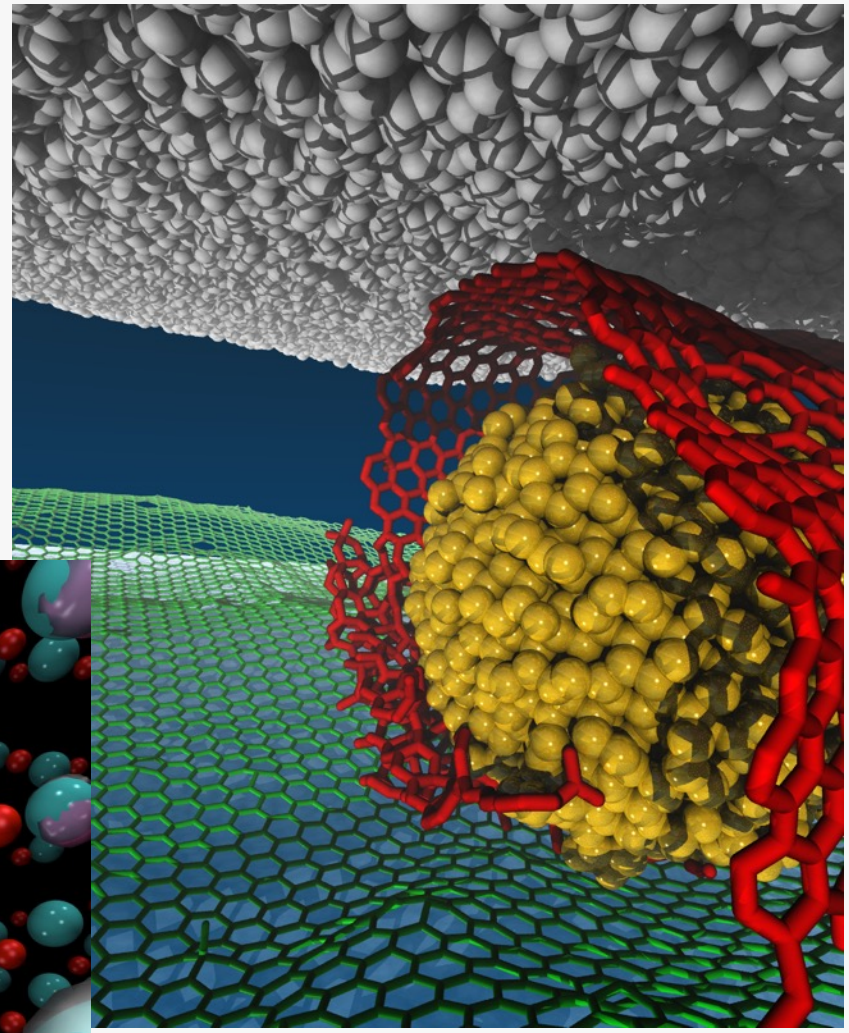
Materials Science / Molecular



Data courtesy of: Jeff Greeley, Nichols Romero, Argonne National Laboratory

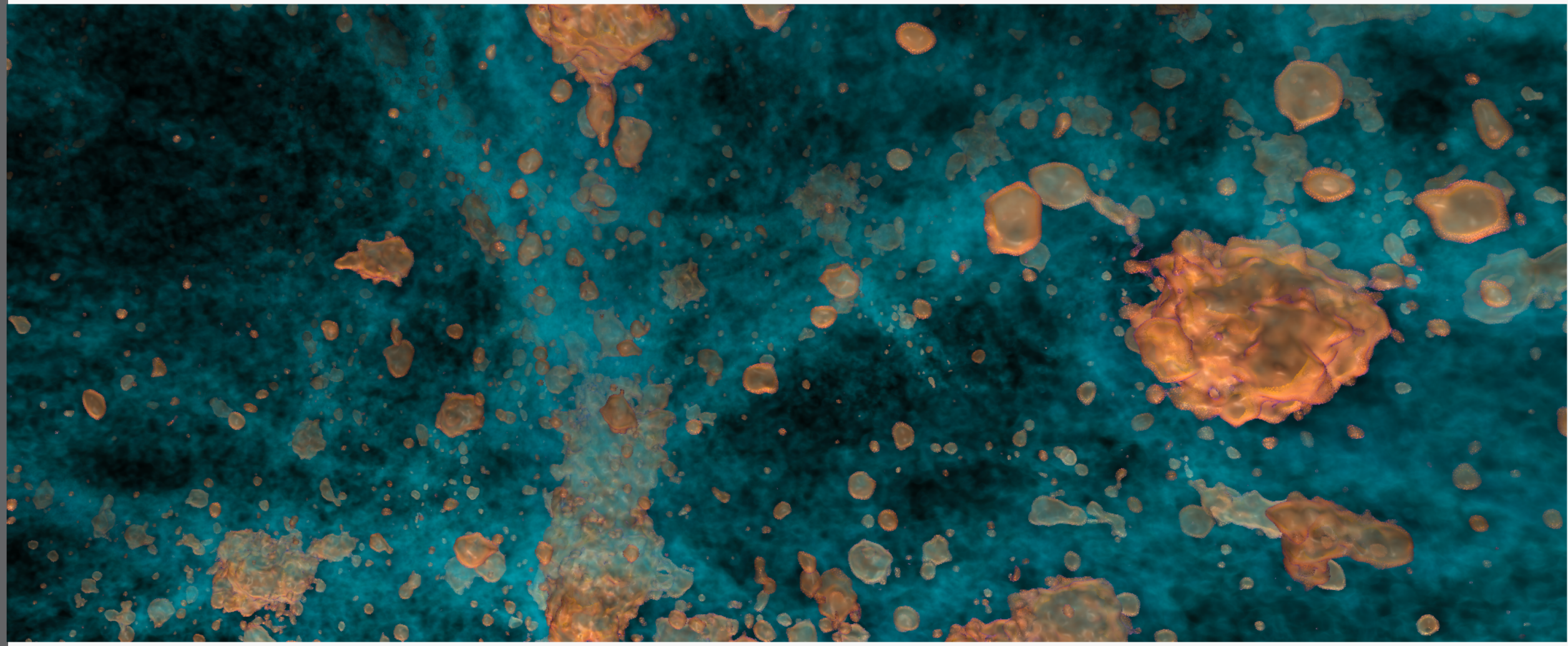
Data courtesy of: Subramanian Sankaranarayanan, Argonne National Laboratory





Data courtesy of: Paul Kent, Oak Ridge National Laboratory, Anouar Benali, Argonne National Laboratory

Cosmology



Data courtesy of: Salman Habib, Katrin Heitmann, and the HACC team, Argonne National Laboratory

Cooley: Analytics/Visualization cluster

Peak 223 TF

126 nodes; each node has

– Two Intel Xeon E5-2620 Haswell 2.4 GHz 6-core processors

– NVIDIA Telsa K80 graphics processing unit (24GB)

– 384 GB of RAM

Aggregate RAM of 47 TB

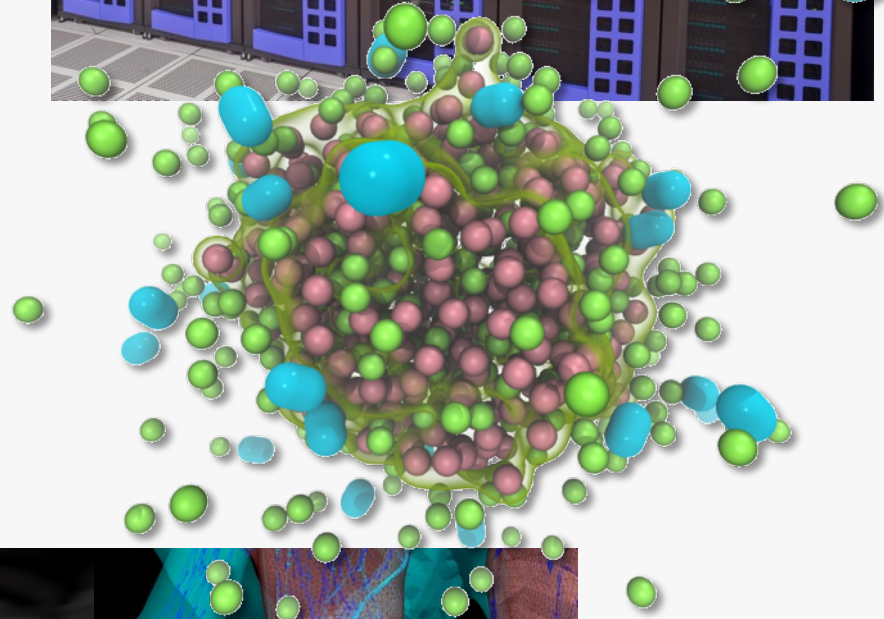
Aggregate GPU memory of ~3TB

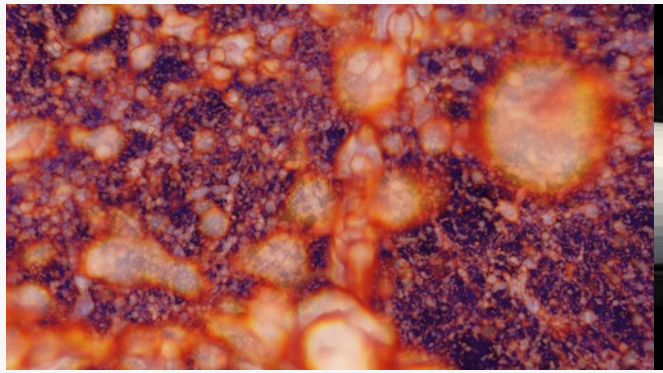
Cray CS System

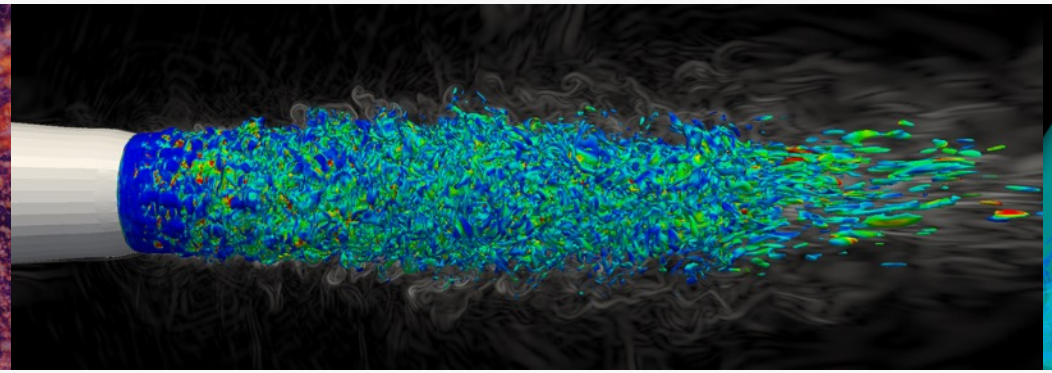
216 port FDR IB switch with uplinks to our QDR infrastructure

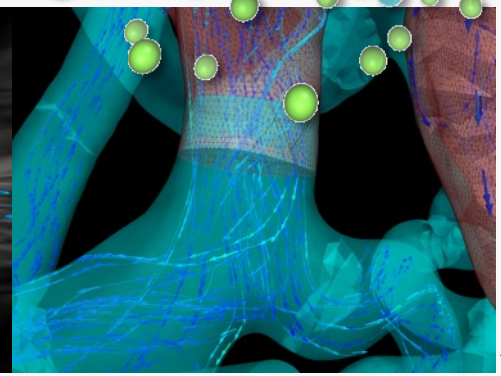
Mounts the Theta file system











Argonne NATIONAL LABORATORY



Visualization Tools and Data Formats

All Sorts of Tools

Visualization Applications
- VisIt *
- ParaView*
- EnSight

Domain Specific
- VMD, PyMol, Ovito

APIs
- VTK*: visualization
- ITK: segmentation & registration

GPU performance
- vl3: shader-based volume and particle rendering

Analysis Environments
- Matlab
- Parallel R

Utilities
- GnuPlot
- ImageMagick*

■ Available on Cooley

* Available on Theta

Argonne NATIONAL LABORATORY

ParaView & VisIt vs. vtk

ParaView & VisIt

- General purpose visualization applications
- GUI-based
- Client / Server model to support remote visualization
- Scriptable / Extendable
- Built on top of vtk (largely)
- *In situ* capabilities

vtk

- Programming environment / API
- Additional capabilities, finer control
- Smaller memory footprint
- Requires more expertise (build custom applications)

Data File Formats (ParaView & VisIt)

VTK

Parallel (partitioned) VTK

VTK MultiBlock (MultiGroup, Hierarchical, Hierarchical Box)

Legacy VTK

Parallel (partitioned) legacy VTK

EnSight files

EnSight Master Server

Exodus

BYU

XDMF

PLOT2D

PLOT3D

SpyPlot CTH

HDF5 raw image data

DEM

VRML

PLY

Polygonal Protein Data Bank

XMol Molecule

Stereo Lithography

Gaussian Cube

Raw (binary)

AVS

Meta Image

Facet

PNG

SAF

LS-Dyna

Nek5000

OVERFLOW

paraDIS

PATRAN

PFLOTRAN

Pixie

PuReMD

S3D

SAS

Tetrad

UNIC

VASP

ZeusMP

ANALYZE

BOV

GMV

Tecplot

Vis5D

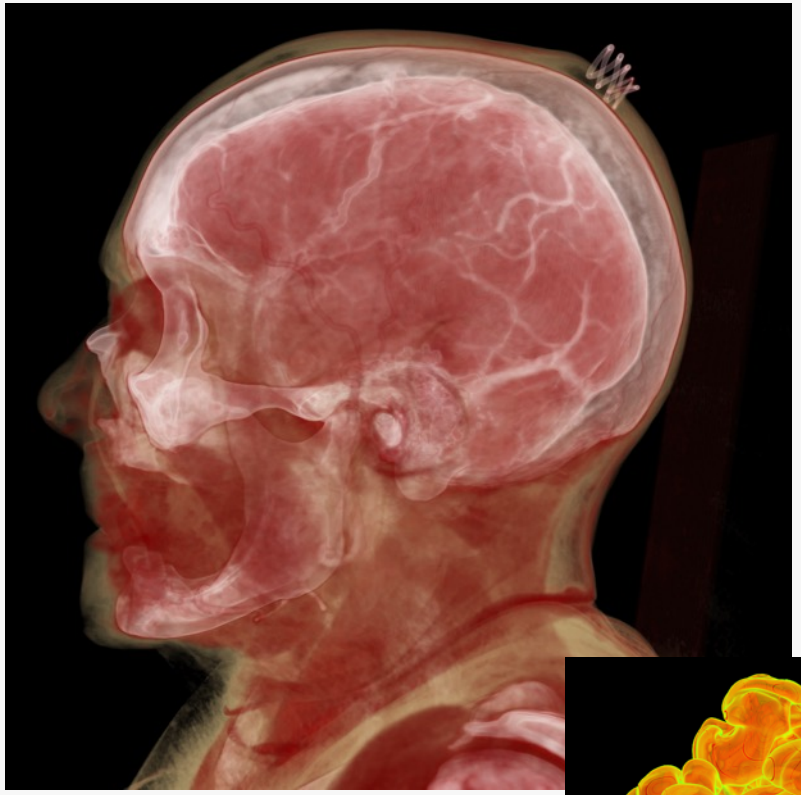
Xmdv

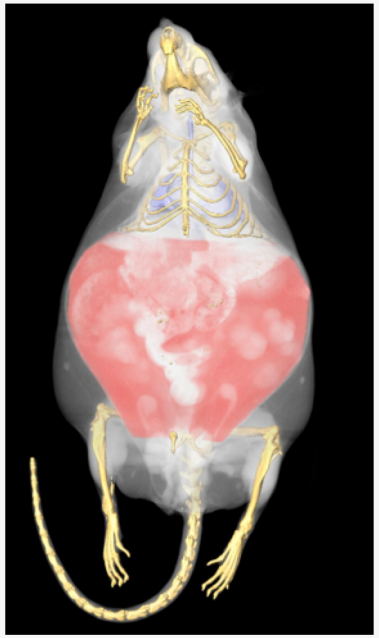
XSF

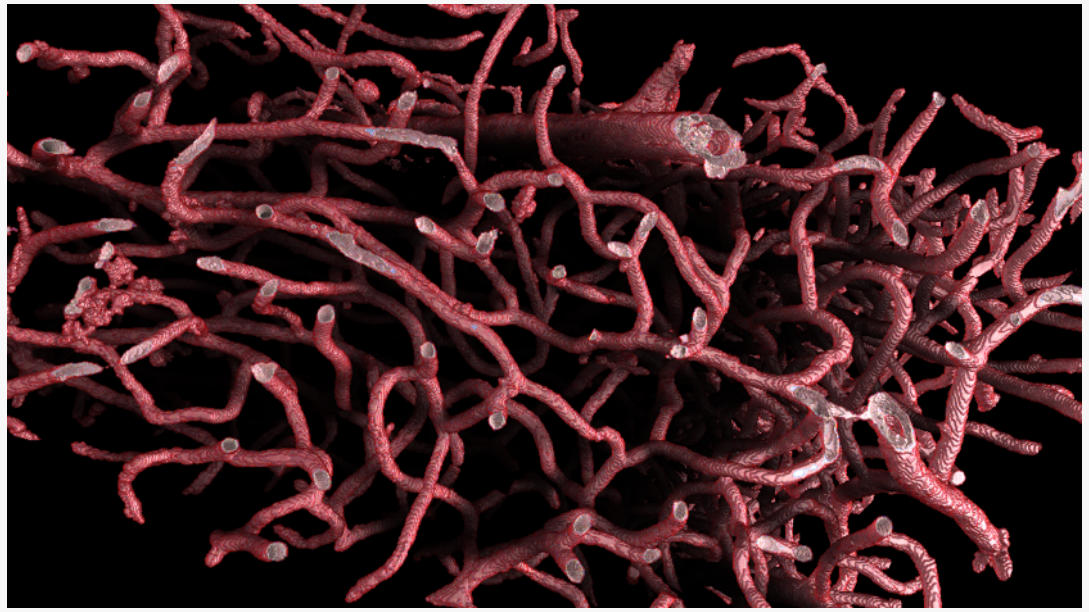


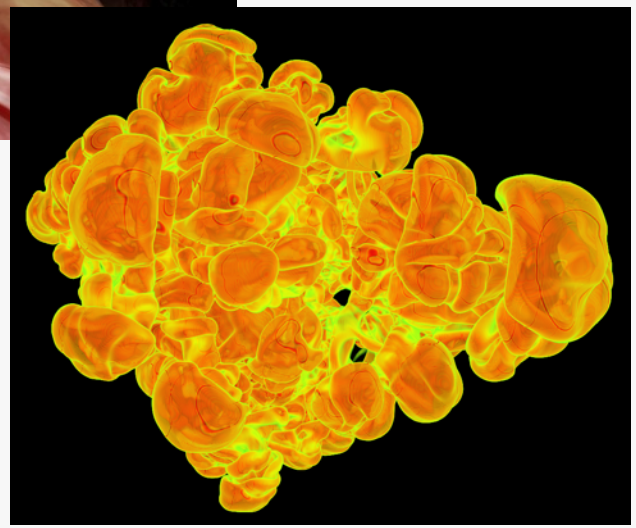
Data Representations

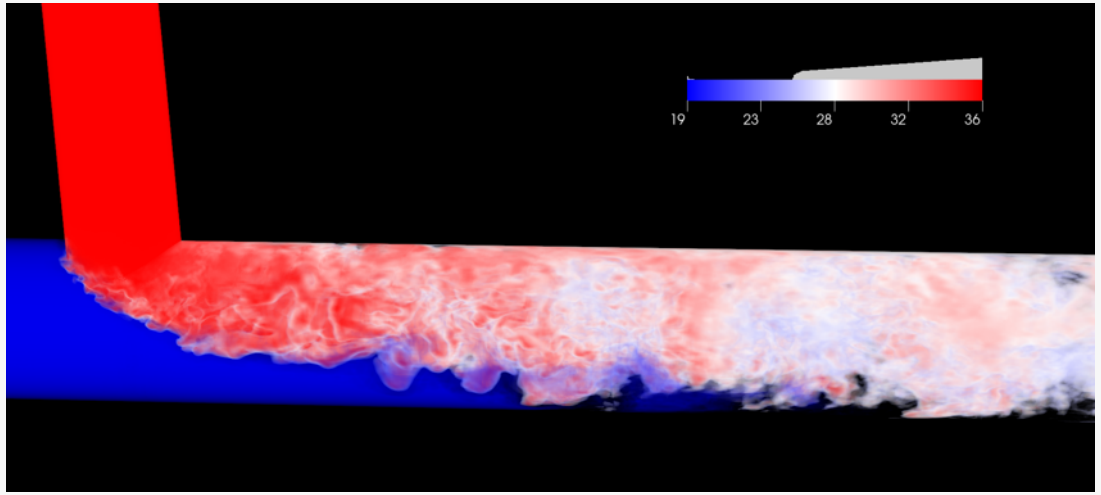
Data Representations: Volume Rendering











Data Representations: Glyphs

2D or 3D geometric object to represent point data

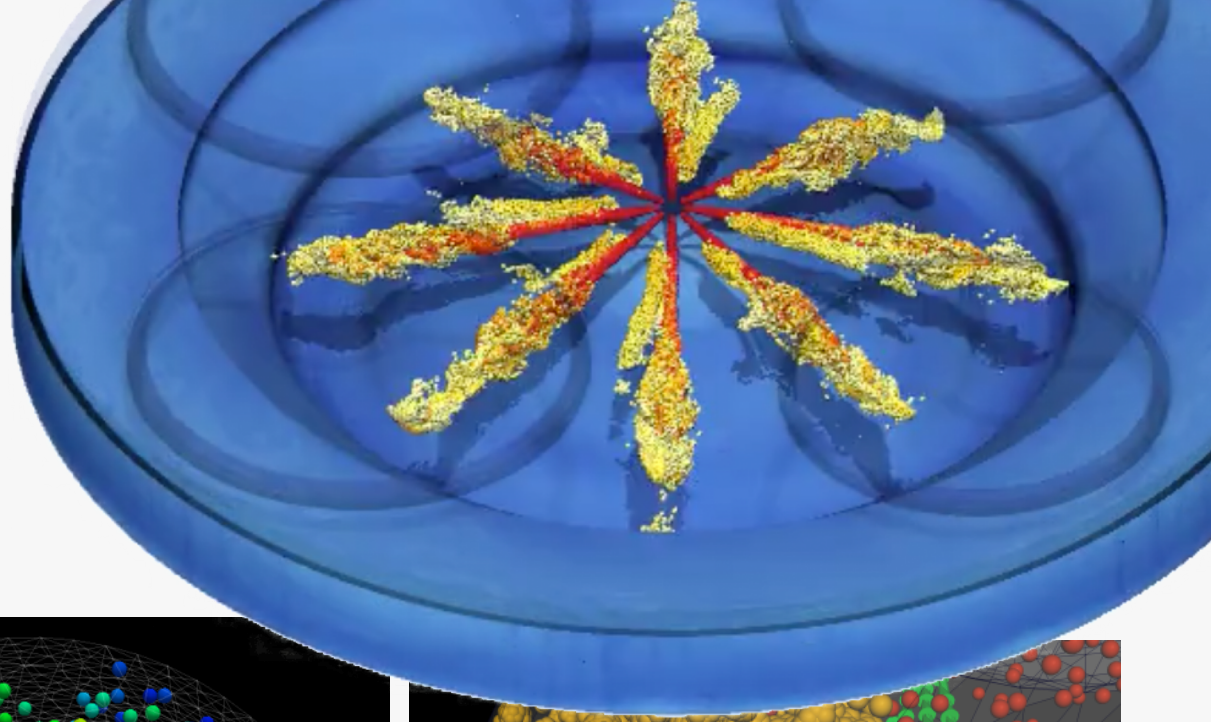
Location dictated by coordinate

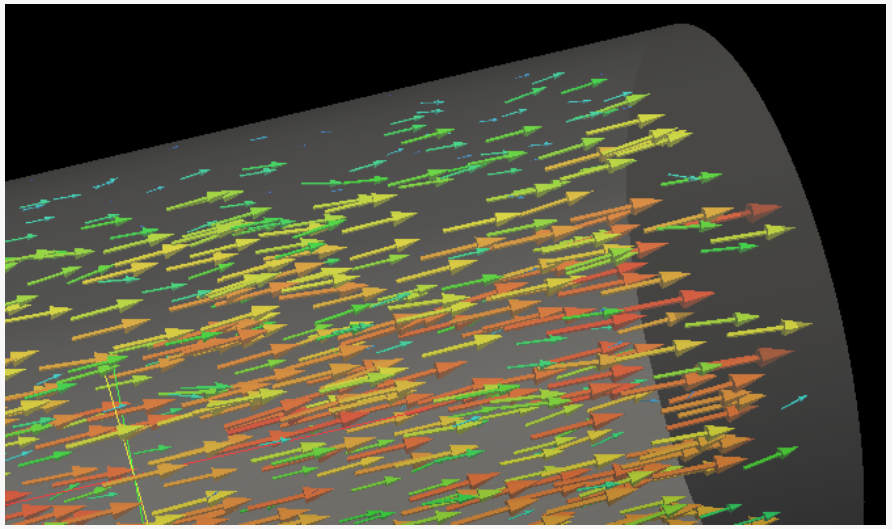
– 3D location on mesh

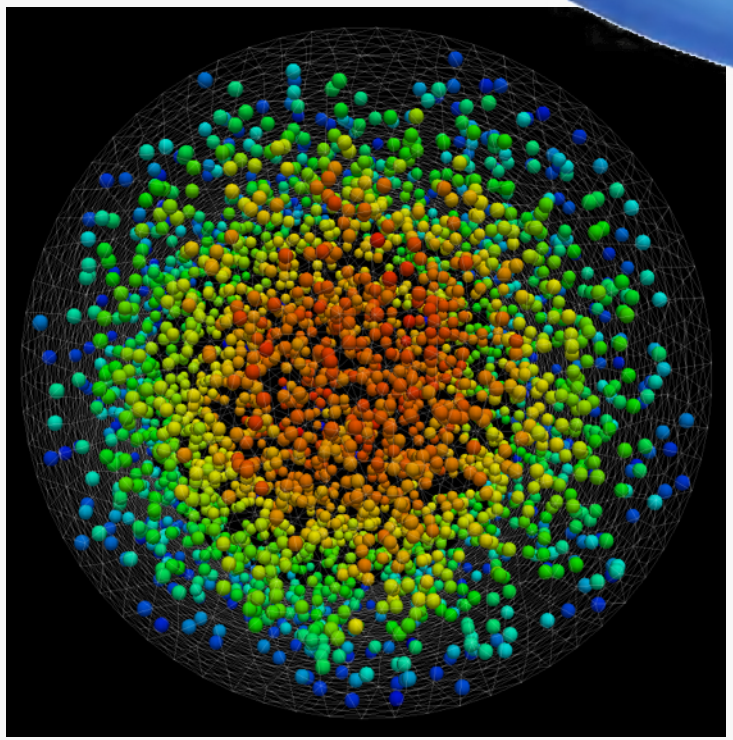
– 2D position in table/graph

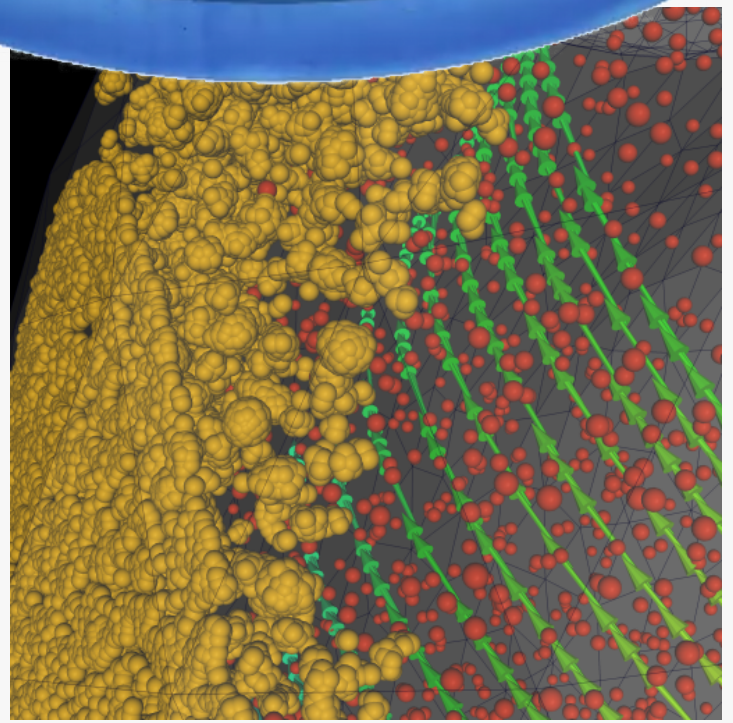
Attributes of graphical entity dictated by attributes of data

– color, size, orientation









Argonne NATIONAL LABORATORY

Data Representations: Contours (Isosurfaces)

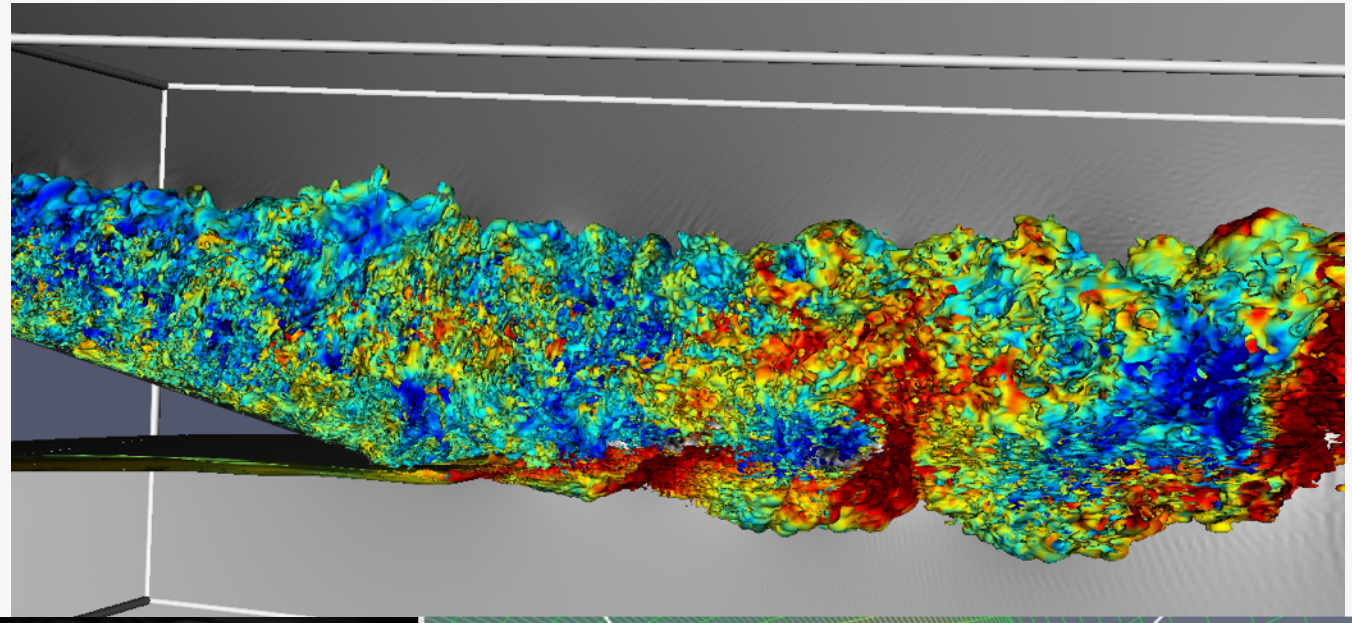
A Line (2D) or Surface (3D), representing a constant value

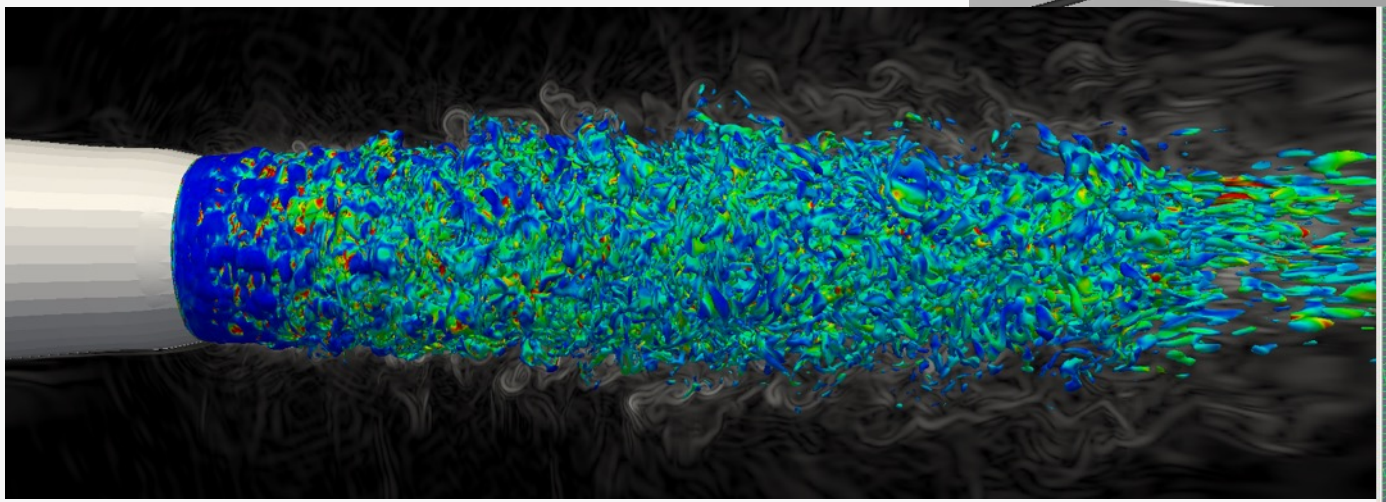
VisIt & ParaView:

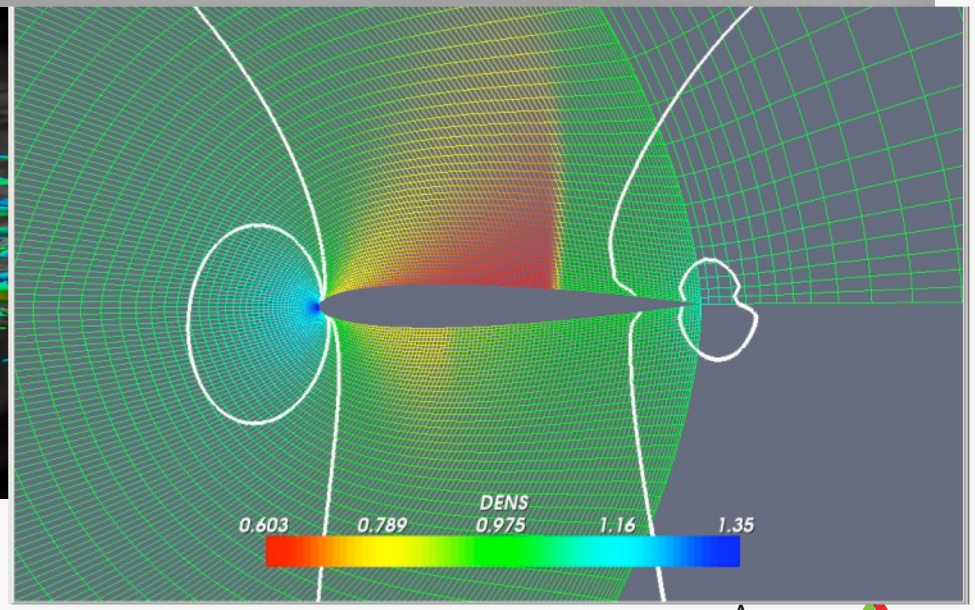
– good at this

vtk:

– same, but again requires more effort







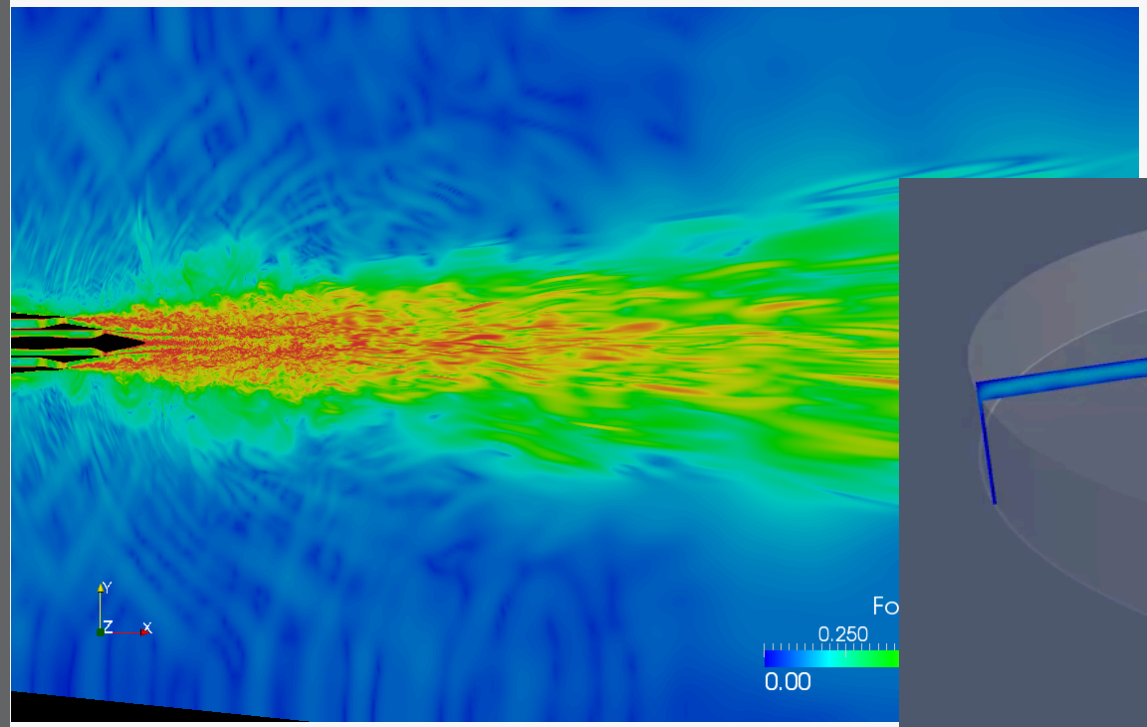
Data Representations: Cutting Planes

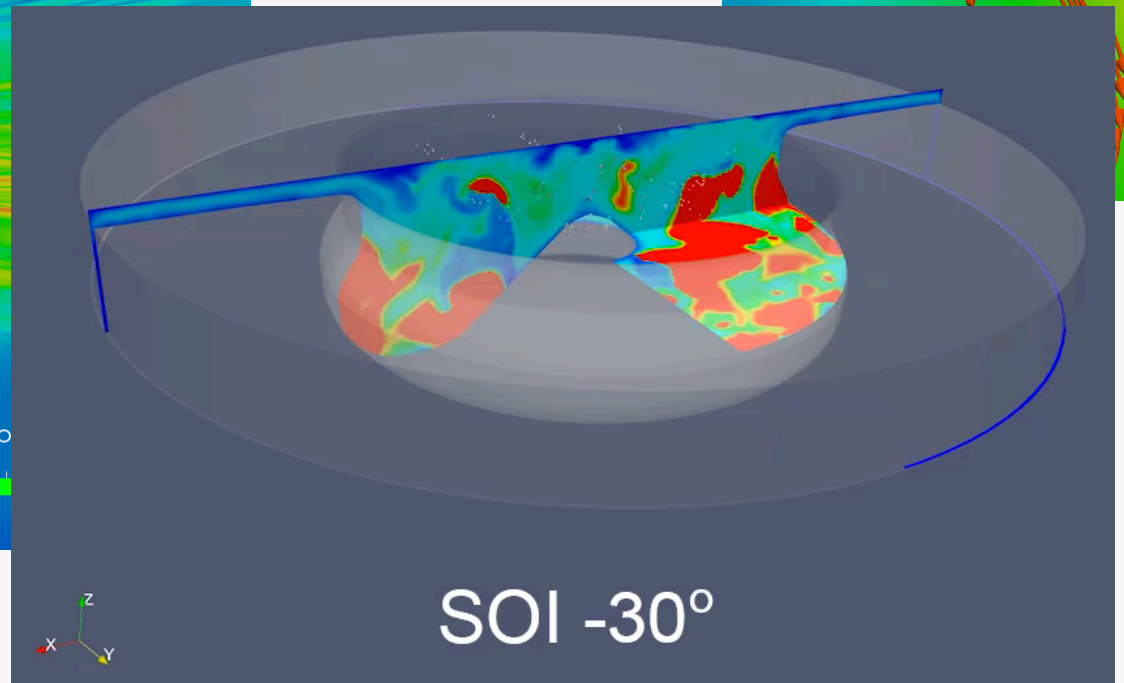
Slice a plane through the data

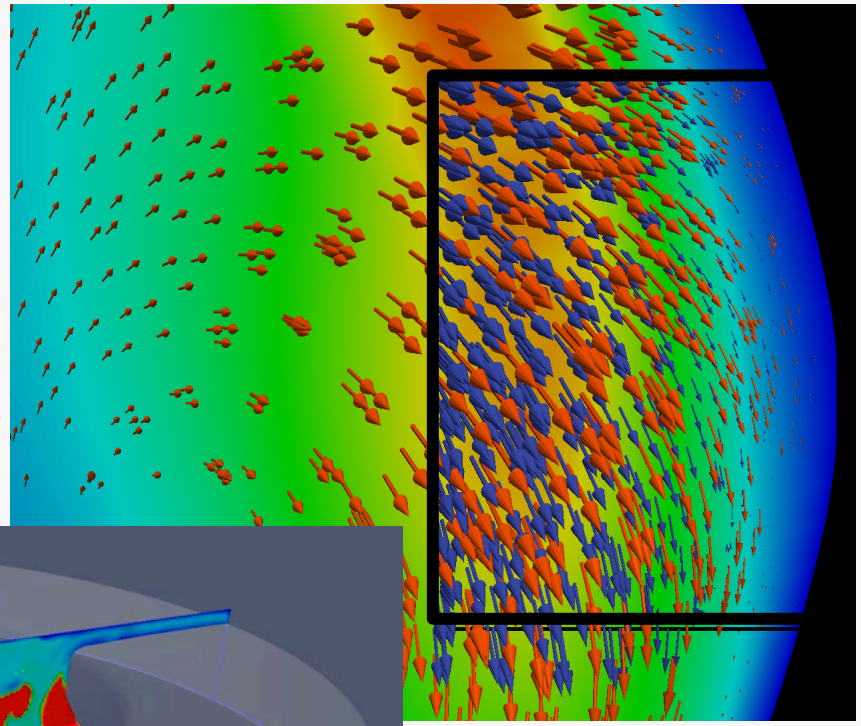
– Can apply additional visualization methods to resulting plane

VisIt & ParaView & vtk good at this

VMD has similar capabilities for some data formats





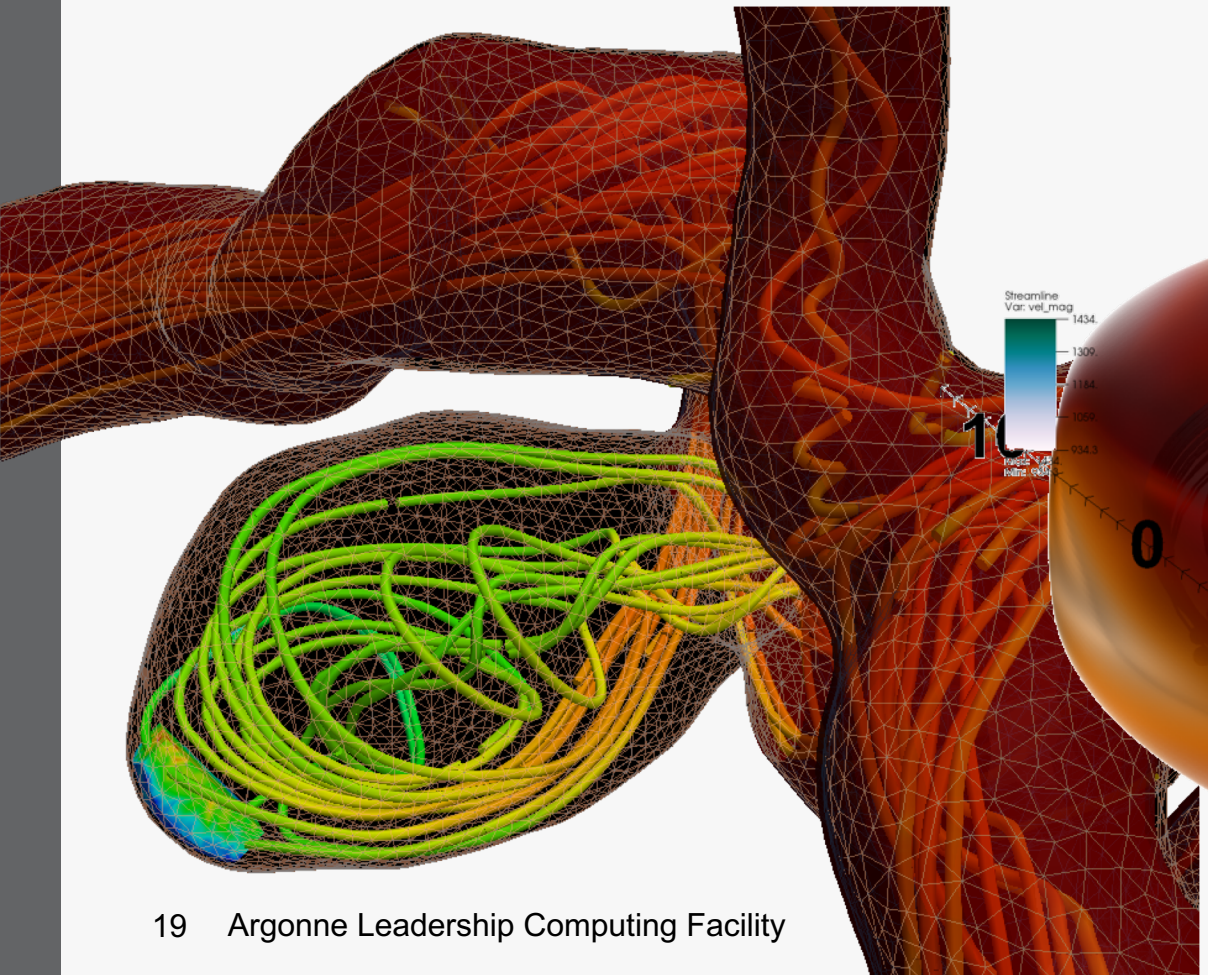


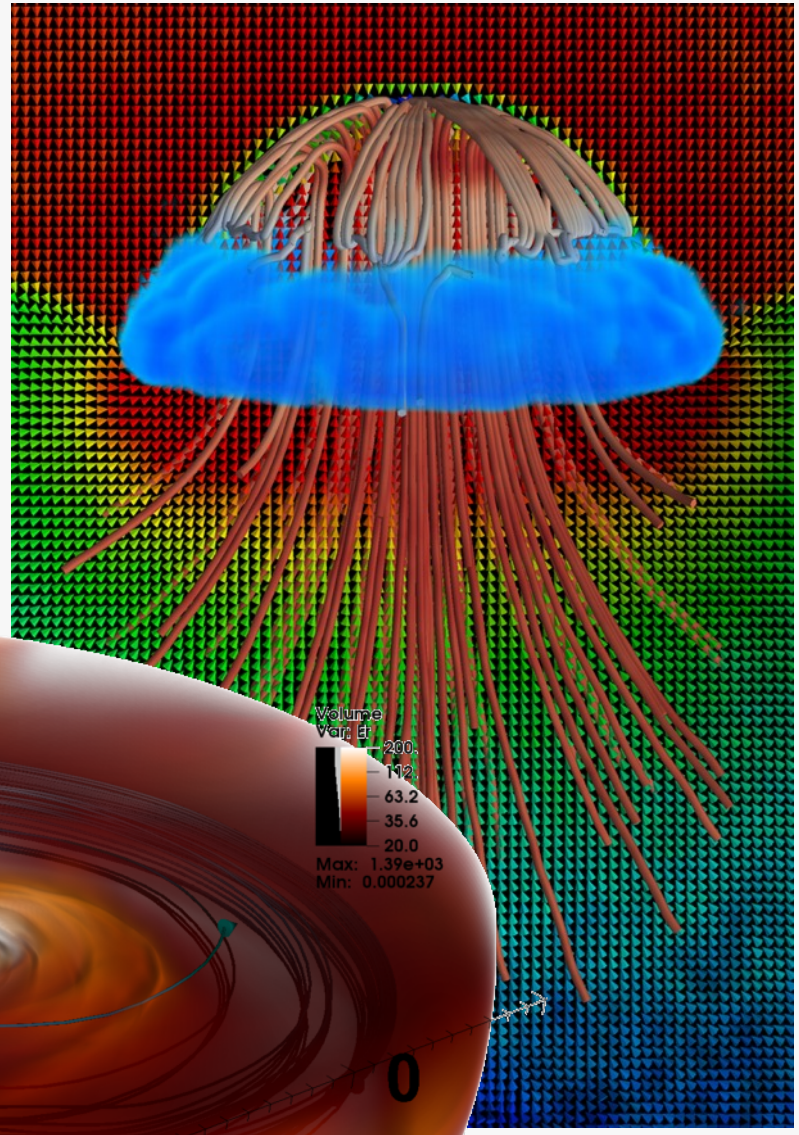
Data Representations: Streamlines

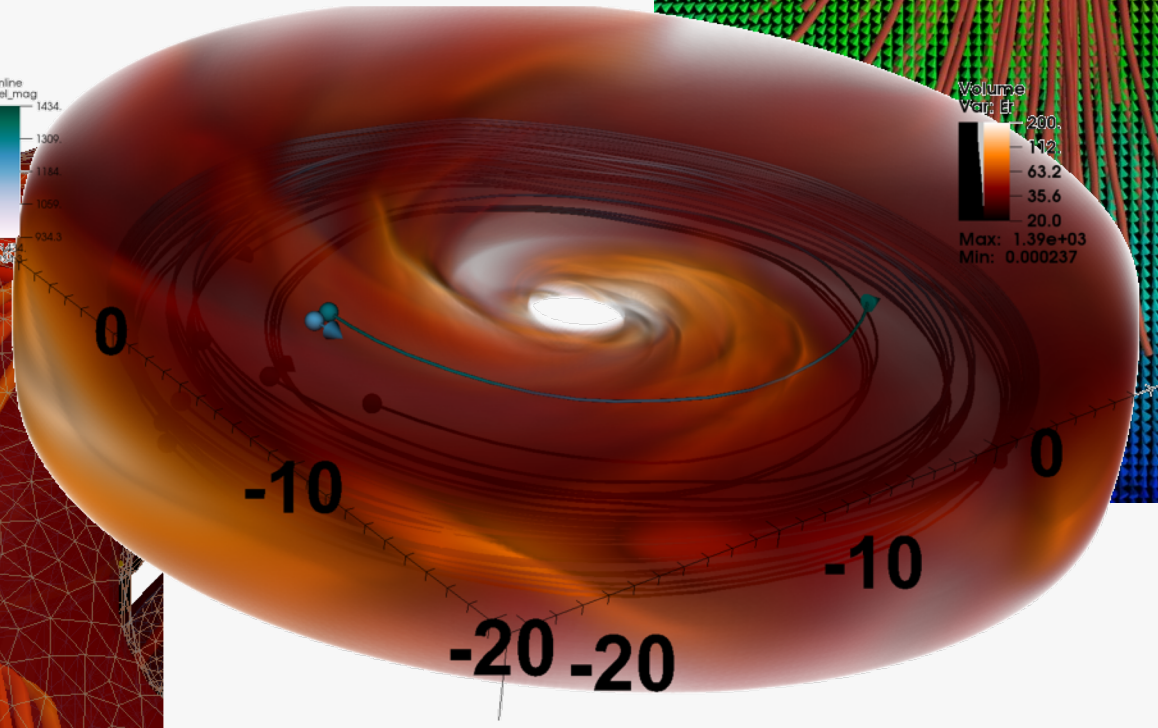
From vector field on a mesh (needs connectivity)

– Show the direction an element will travel in at any point in time.

VisIt & ParaView & vtk good at this







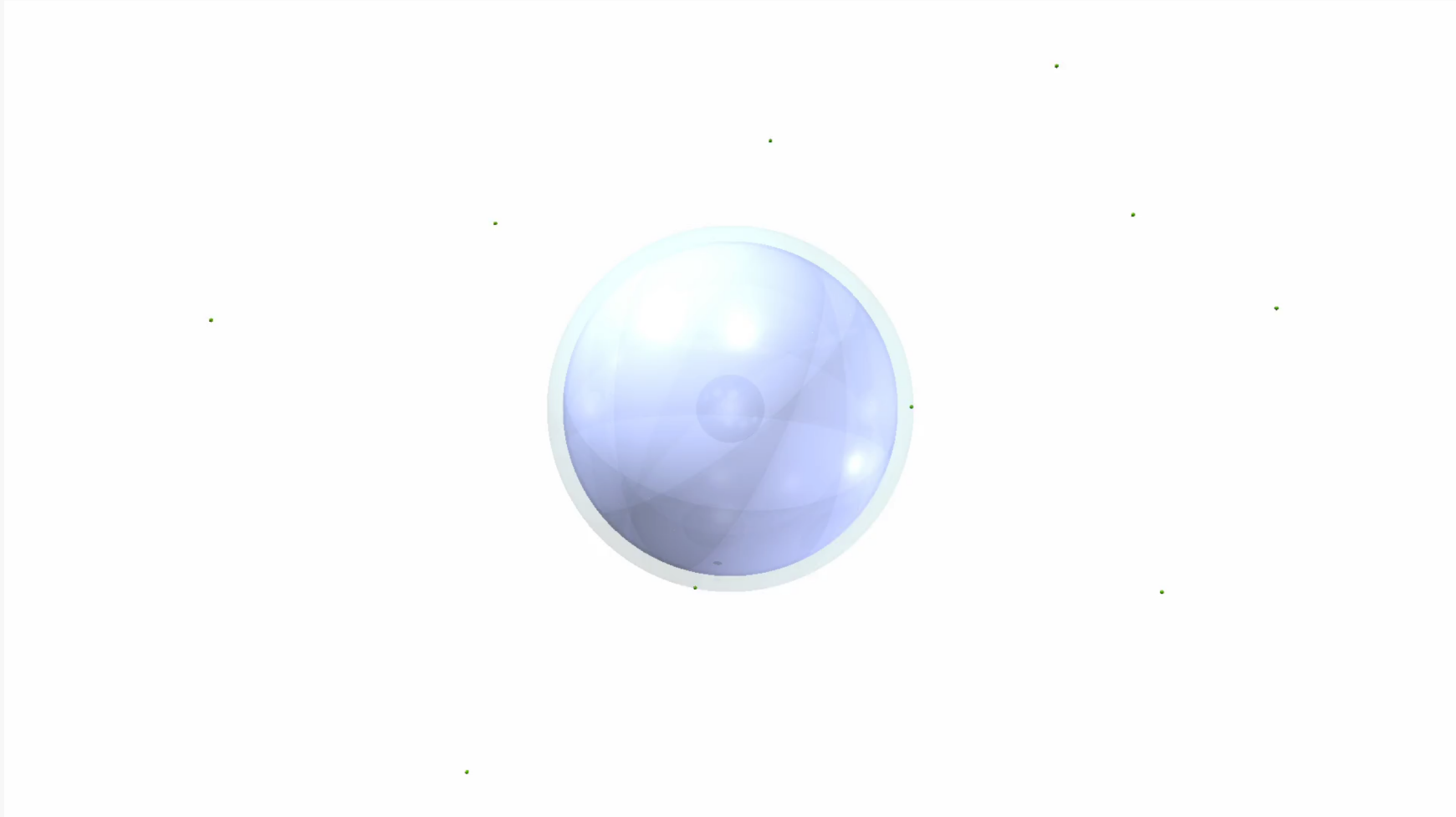
Argonne NATIONAL LABORATORY

Data Representations: Pathlines

From vector field on a mesh (needs connectivity)

– Trace the path an element will travel over time.

VisIt & ParaView & vtk good at this



Molecular Dynamics Visualization

VMD:

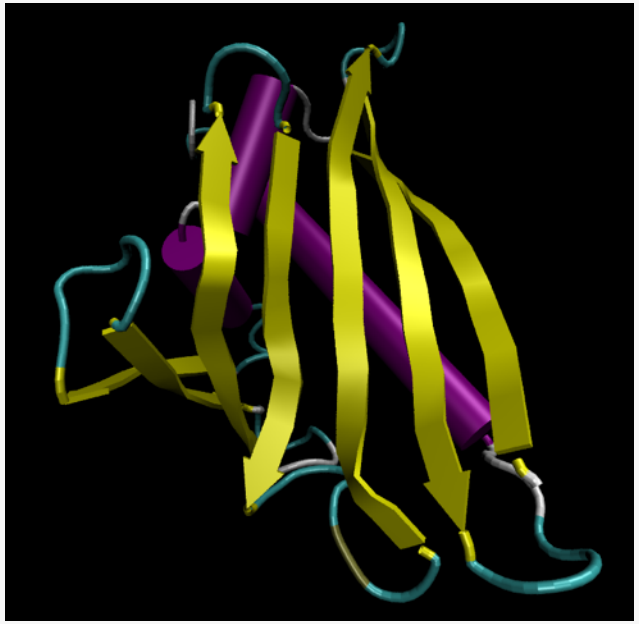
– Lots of domain-specific representations
– Many different file formats
– Animation
– Scriptable

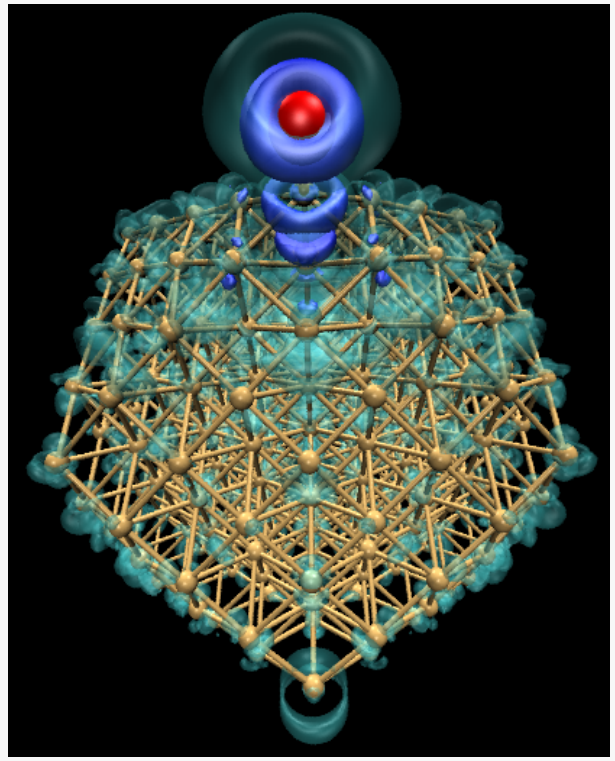
VisIt & ParaView:

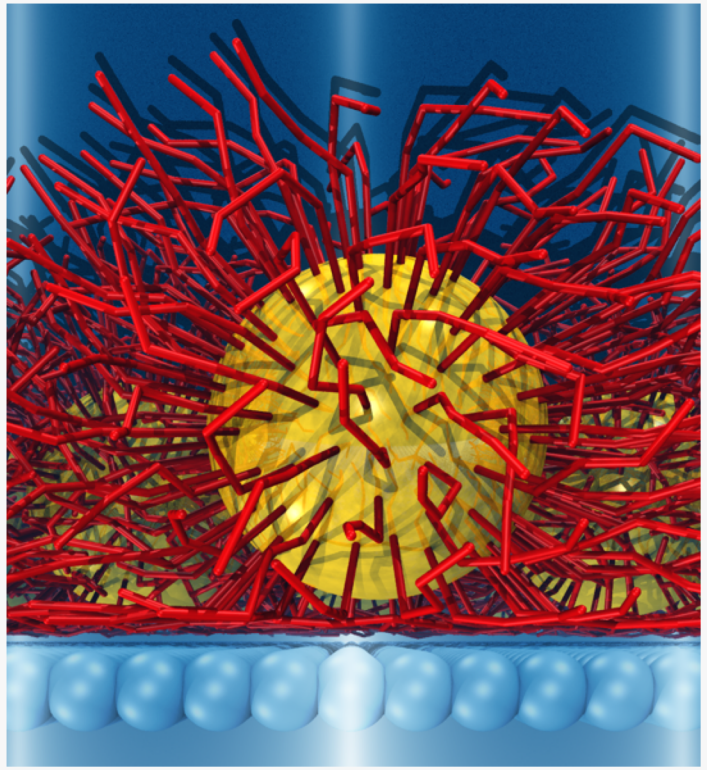
– Limited support for these types of representations, but improving

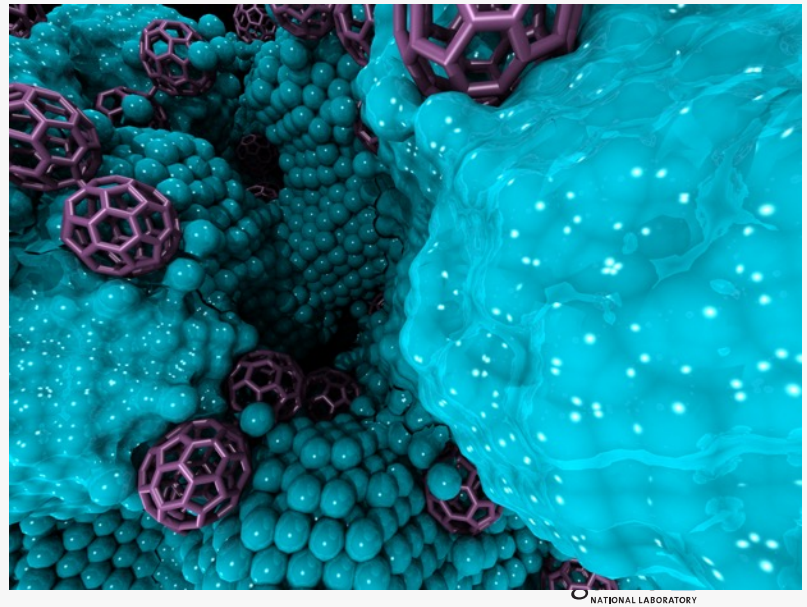
VTK:

– Anything's possible if you try hard enough



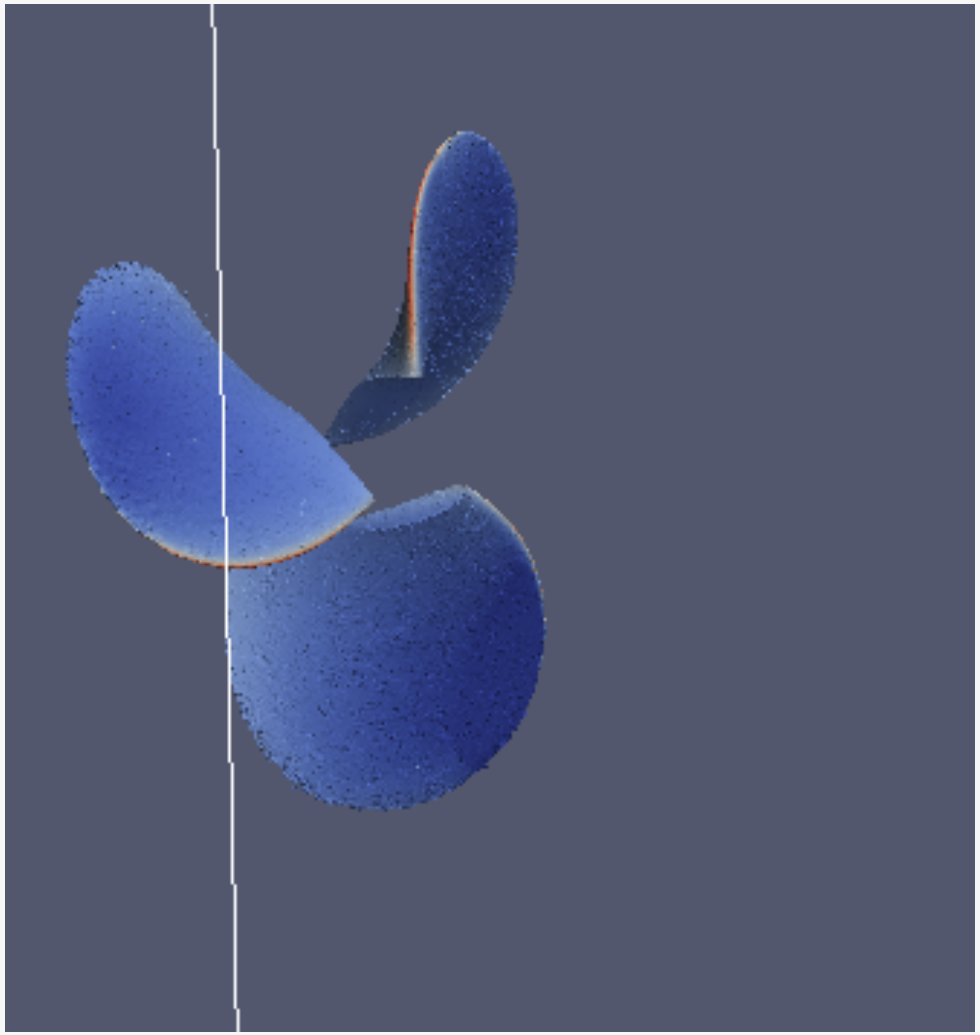






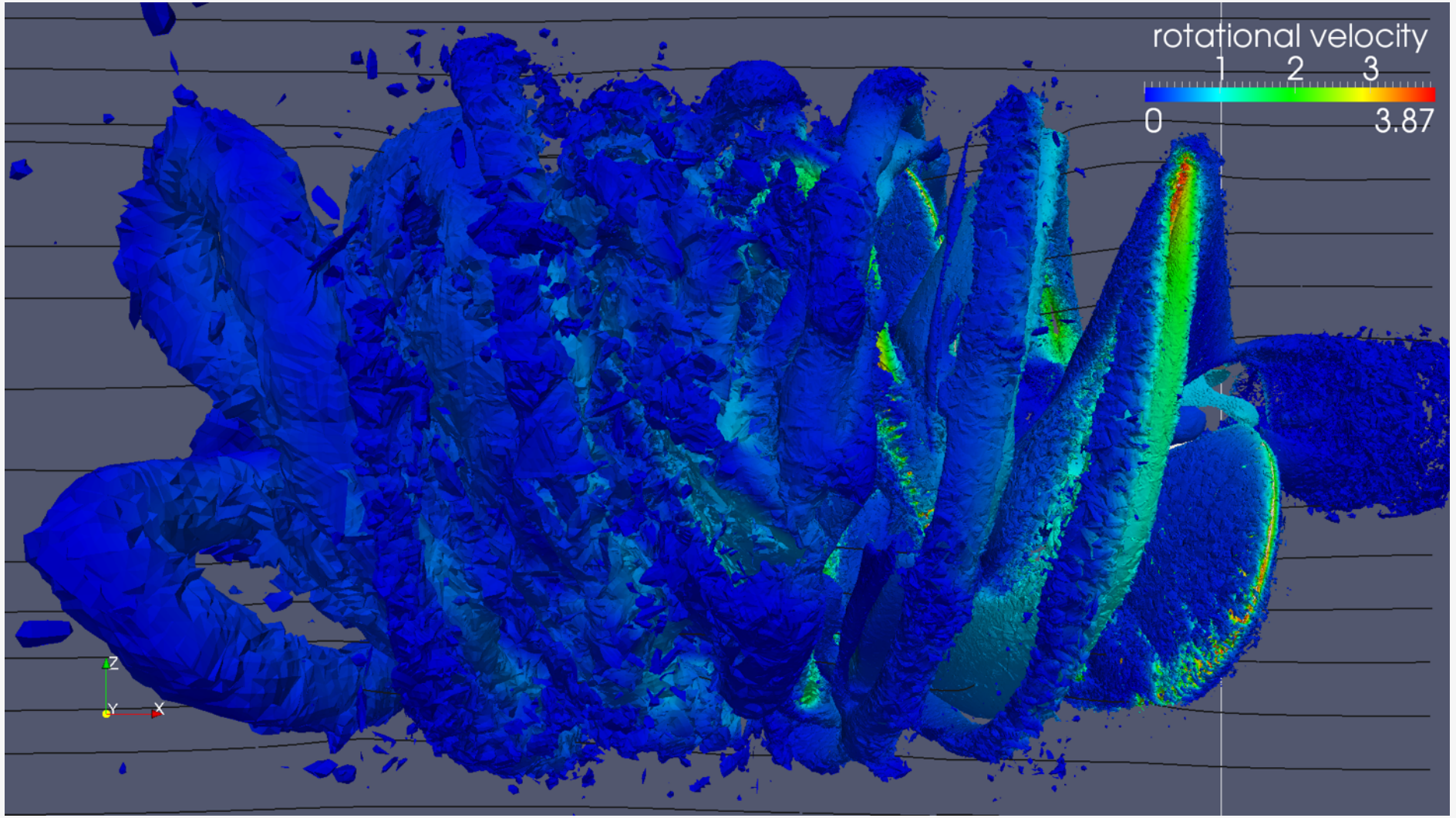
Visualization for Debugging

Visualization for Debugging

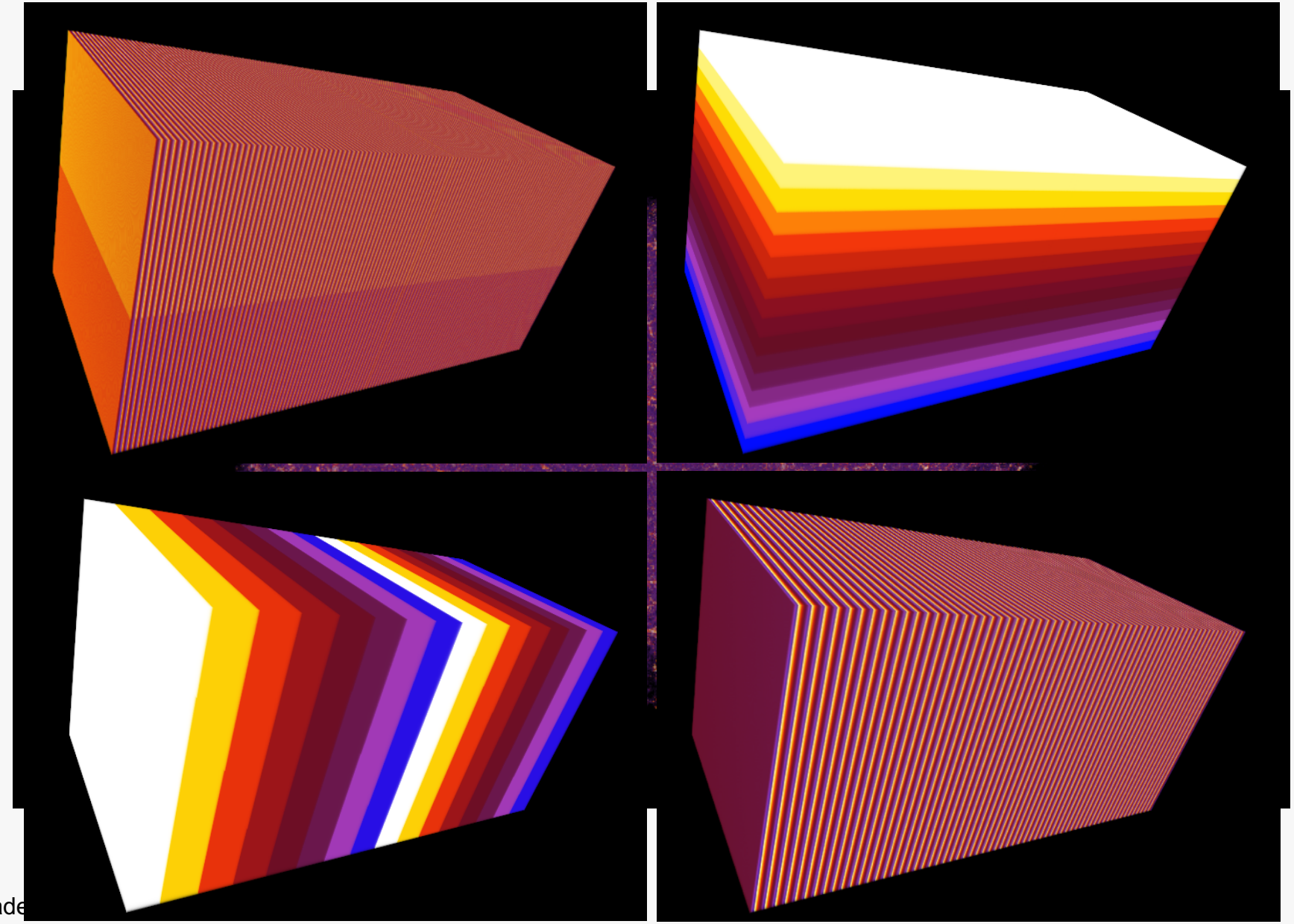


Argonne NATIONAL LABORATORY

Visualization for Debugging

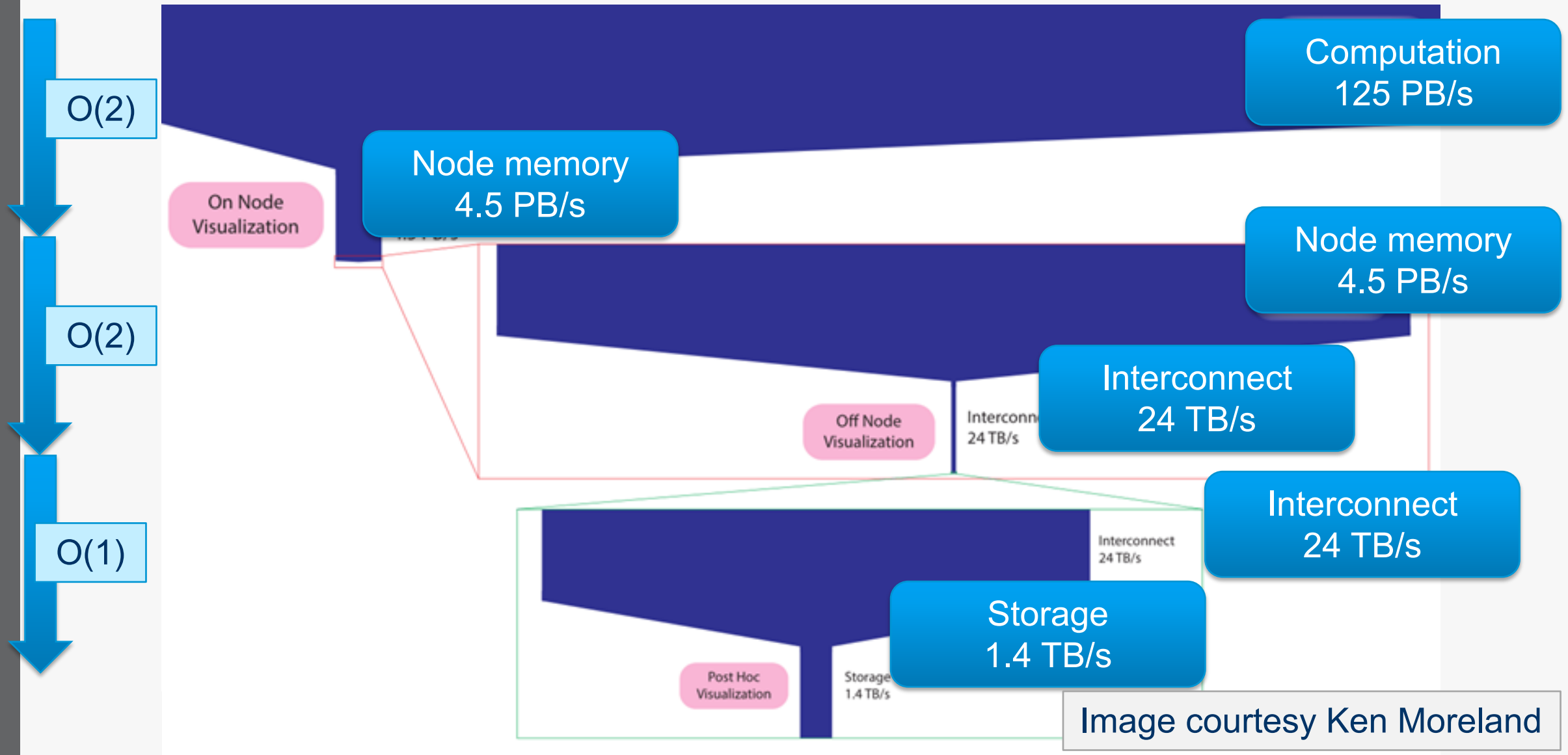


Visualization as Diagnostics: Color by Thread ID



In Situ Visualization and Analysis

Five orders of magnitude between compute and I/O capacity on Titan Cray system at ORNL

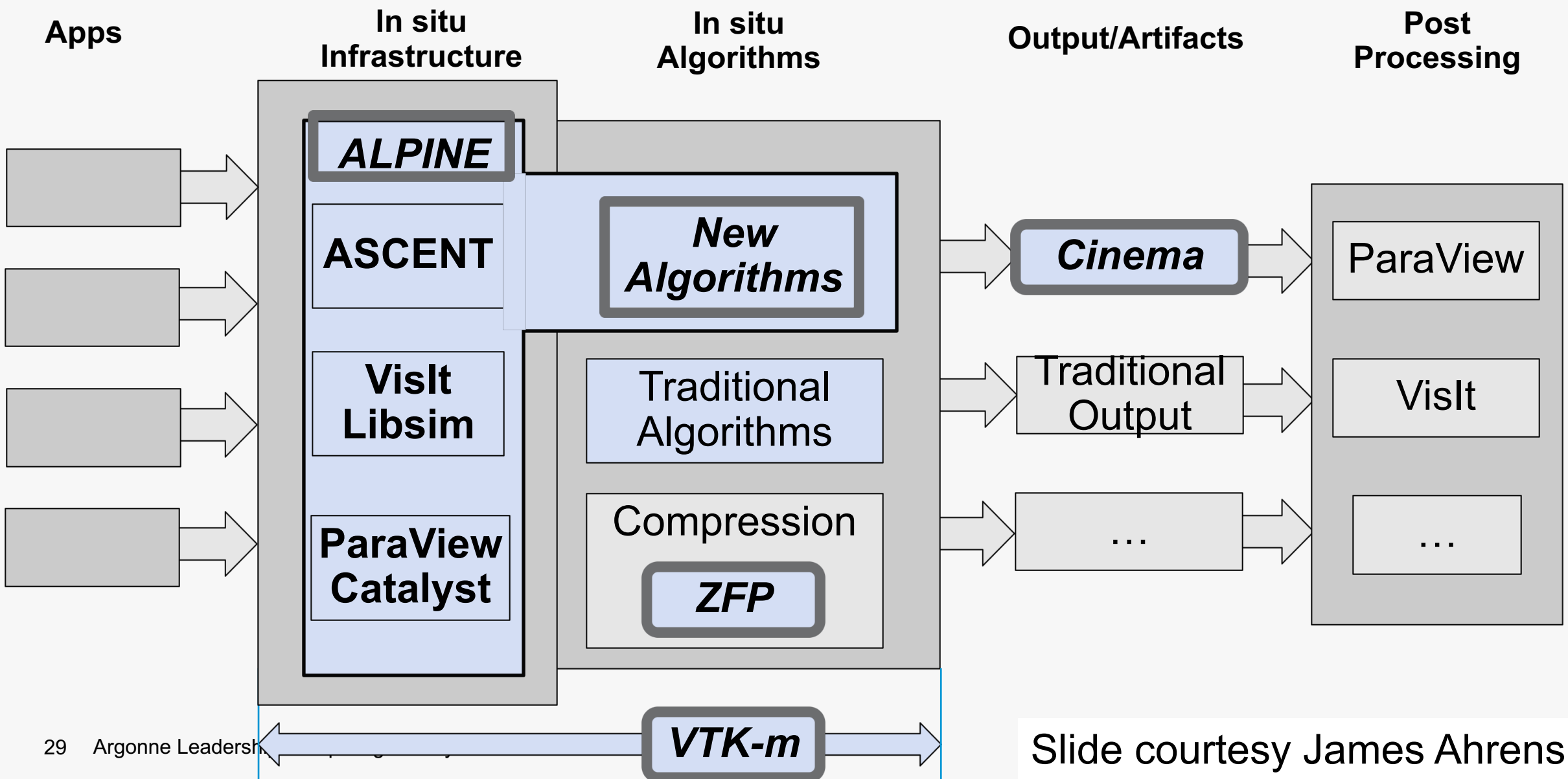


What are the problems?

- Not enough I/O capacity on current HPC systems, and the trend is getting worse.
- If there's not enough I/O, you can't write data to storage, so you can't analyze it: lost science.
- Energy consumption: it costs a lot of power to write data to disk.
- Opportunity for doing better science (analysis) when have access to full spatiotemporal resolution data.

Slide courtesy the SENSEI team

ECP Software Technology Data and Visualization projects provide an integrated workflow



Slide courtesy James Ahrens

Cinema

Approach: Visualize all results needed while simulation data is in memory

– Operators, camera positions/angles, simulation and algorithmic parameters

Result: Database of images

– Pixel accurate results when compared to post-processing rendering of the same data

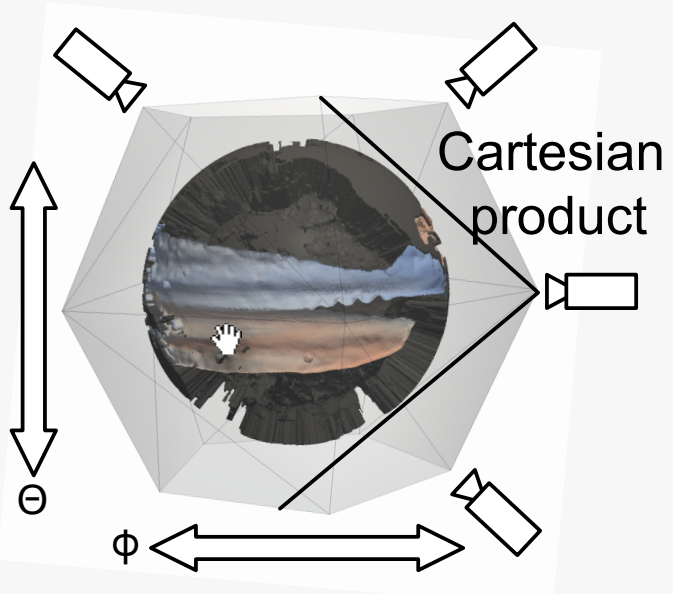
Properties of this solution:

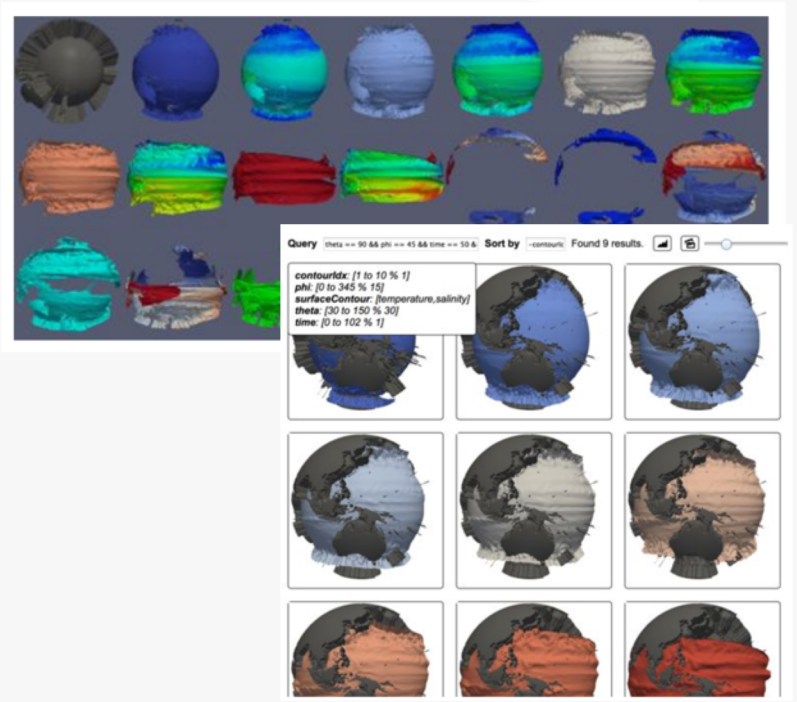
– Sampling of visualization result image output space

– Visualization/rendering as sampling/data reduction operator

– Note we can evaluate and optimize this finite set of image outputs

Cinemascience.org





Slide courtesy James Ahrens and the Cinema team

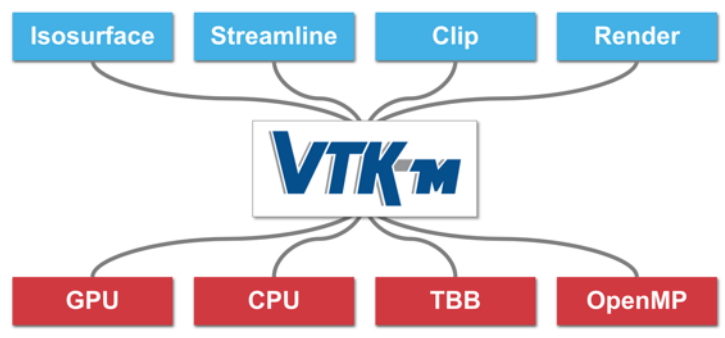
VTK-m Delivering Portable Performance of Visualization Algorithms Across ECP Processors

Development is accelerated by VTK-m's write once, run everywhere

– Data parallel programming abstraction

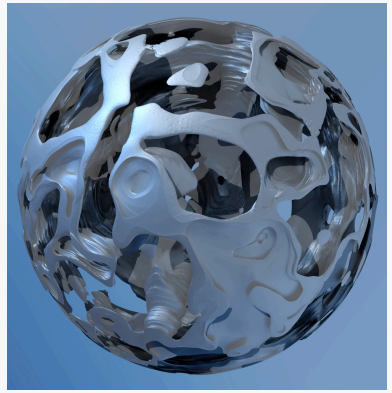
Ray tracing performance in VTK-m is within a factor of 2 of highly optimized, processor-specific implementations developed by Intel and NVIDIA [Moreland, et al., 2016]

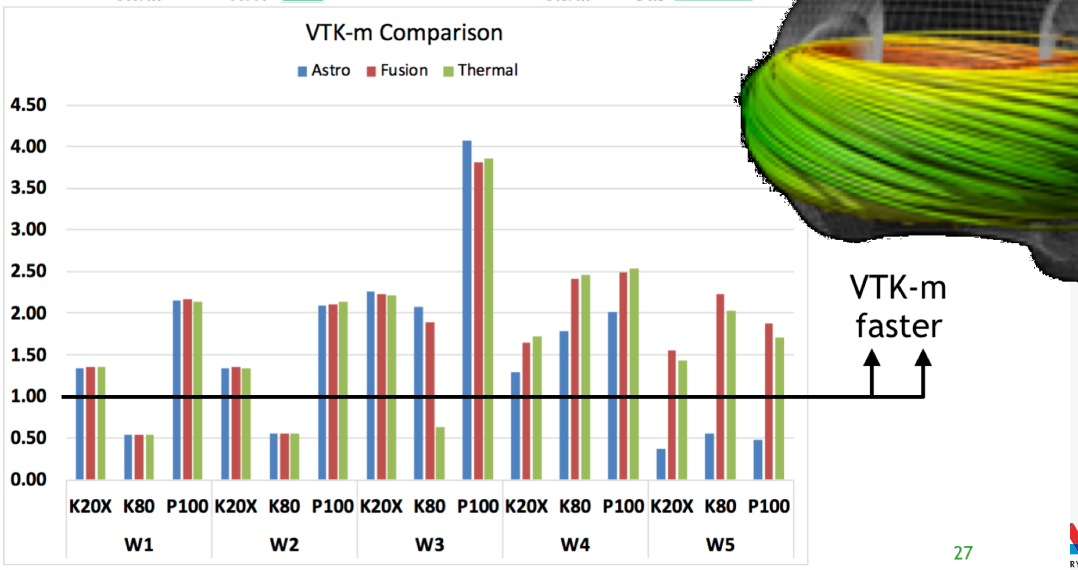
Particle advection in VTK-m performs *better* than code specifically developed for CUDA processors [Pugmire, et al. 2018]

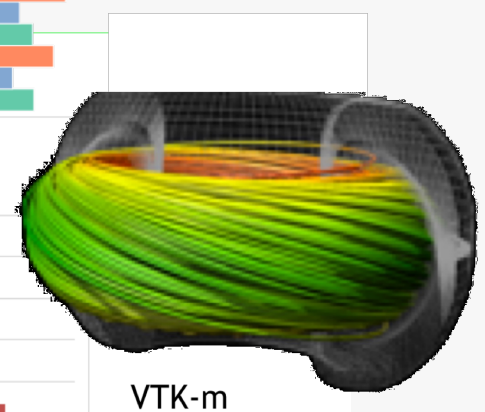


Dataset	Algorithm	Millions of rays per second
LT_350K	OptiX Prime	357.6
	EAVL	150.8
	VTK-m	164.5
LT_372K	OptiX Prime	322.4
	EAVL	124.7
	VTK-m	140.8
RM_350K	OptiX Prime	436.5
	EAVL	197.5
	VTK-m	200.8
RM_650K	OptiX Prime	420.4
	EAVL	172.9
	VTK-m	166.0
RM_970K	OptiX Prime	347.1
	EAVL	152.8
	VTK-m	163.5
RM_1.7M	OptiX Prime	266.8
	EAVL	136.6
	VTK-m	148.8
RM_3.2M	OptiX Prime	264.5
	EAVL	124.8
	VTK-m	134.5
Seismic	OptiX Prime	267.8
	EAVL	106.3
	VTK-m	119.4

Dataset	Algorithm	Millions of rays per second
LT_350K	Embree	51.9
	EAVL	27.7
	VTK-m	38.5
LT_372K	Embree	56.5
	EAVL	26.1
	VTK-m	36.0
RM_350K	Embree	64.8
	EAVL	33.3
	VTK-m	47.8
RM_650K	Embree	65.9
	EAVL	35.6
	VTK-m	49.1
RM_970K	Embree	59.1
	EAVL	29.3
	VTK-m	41.0
RM_1.7M	Embree	52.4
	EAVL	27.0
	VTK-m	37.8
RM_3.2M	Embree	48.4
	EAVL	28.3
	VTK-m	33.9
Seismic	Embree	43.2
	EAVL	25.2
	VTK-m	34.5







Slide courtesy James Ahrens and the VTK-m team

Two Frameworks for In Situ Vis and Analysis at ALCF





- "Write once, run everywhere" design
- Data model based on VTK from Kitware
- Supports a variety of backends, including ParaView/Catalyst, VisIt/LibSim, ADIOS, Python

- Flyweight design, minimizes dependencies
- Data model based on Conduit from LLNL
- Vis and analysis algorithms implemented in VTK-m

Instrumenting Simulation Codes





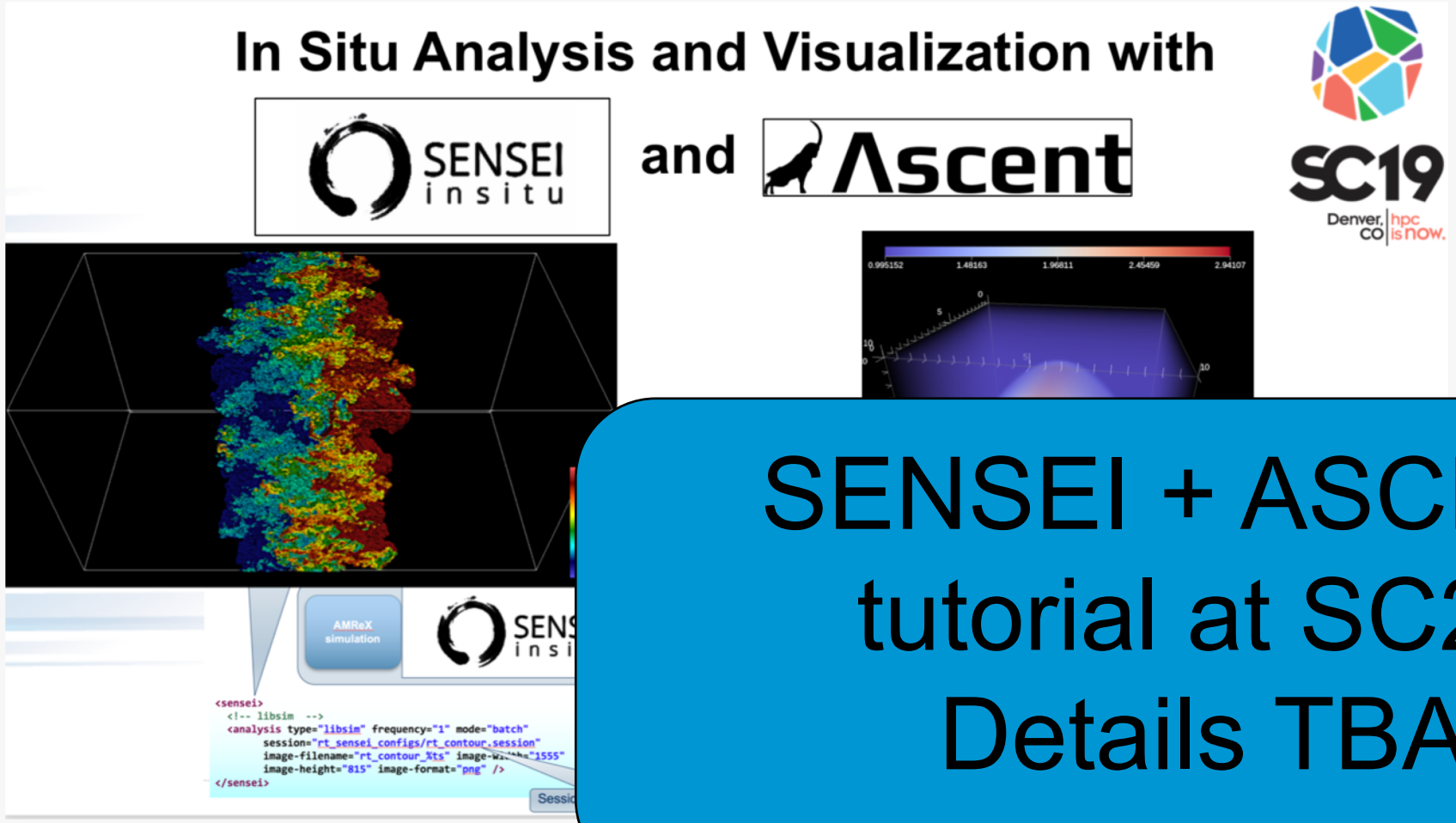

```
1.  initialize sim
2.  if do_insitu bridge::initialize
3.  do
4.    compute new state
5.    if do_io write plot file
6.    if do_insitu bridge::execute
7.  while !done
8.  if do_insitu bridge::finalize
9.  finalize sim
```

```
//
// Run Ascent
//

Ascent ascent;
ascent.open();
ascent.publish(data);
ascent.execute(actions);
ascent.close();
```

SENSEI + ASCENT tutorial at SC19

Slides and Virtual Machine available here:
https://sensei-insitu.org/tutorials/sc19.html



SENSEI + ASCENT tutorial at SC20
Details TBA

Additional Resources

Visualization Help for ALCF users

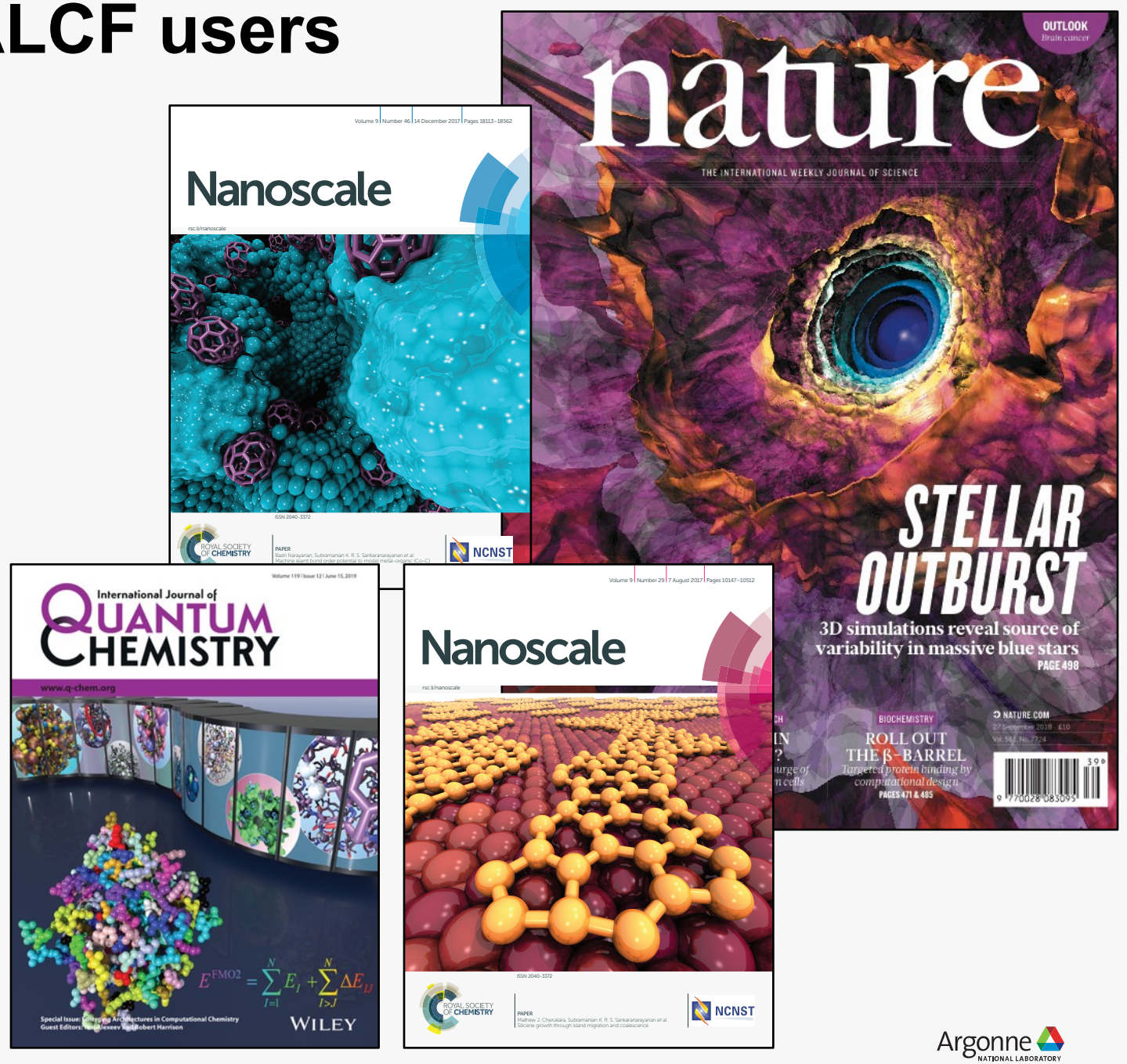
support@alcf.anl.gov

Publication Images & Covers

Animations

- SC Visualization Showcase [Best Vis Finalist 2014-2019]
- APS Division of Fluid Dynamics Gallery of Fluid Motion
- SC Gordon Bell Submissions
- Press Releases

InSitu Vis and Analysis



In situ additional resources



- Project page
  https://sensei-insitu.org/
- Repository
  https://gitlab.kitware.com/sensei/sensei



- Website + Docs:
  http://ascent-dav.org
- Repository:
  https://github.com/Alpine-DAV/ascent

2019 ASCR Workshop on In Situ Data Management

- **Pervasive ISDM**: Apply ISDM methodologies and in situ workflows at a variety of platforms and scales.
- **Co-designed ISDM**: Coordinate the development of ISDM with the underlying system software so that it is part of the software stack.
- **In Situ Algorithms**: Redesign data analysis algorithms for the in situ paradigm.
- **Controllable ISDM**: Understand the design space of autonomous decision-making and control of in situ workflows.
- **Composable ISDM**: Develop interoperable ISDM components and capabilities for an agile and sustainable programming paradigm.
- **Transparent ISDM**: Increase confidence in reproducible science, deliver repeatable performance, and discover new data features through the provenance of ISDM.

https://www.osti.gov/servlets/purl/1493245



QUESTIONS?

Joe Insley
insley@anl.gov

Janet Knowles
jknowles@anl.gov

Silvio Rizzi
srizzi@anl.gov